

CORTEx

Version 1.0 User's Manual



 **Motion Analysis**

Motion Analysis Corporation: Software License Agreement Terms and Conditions

Definitions

The following terms are defined for the purpose of this Agreement as follows:

(a) "Designated System" means the specified computer system described on the facing of this agreement, which Licensee has purchased from the Licensor.

(b) "Licensed Program" means the software program described on the facing page of this Agreement in object code form only, any updates subsequently provided by License, all permitted copies made by Licensee, and all basic or related materials pertinent to such programs.

License

Under a license granted under this Agreement, License is authorized, on a non-exclusive basis, to use the Licensed Program on the Designated System. License shall refrain from taking any action, such as reverse assembly or reverse compilation, to derive a source code equivalent of the Licensed Program. A license shall be valid until terminated under this Agreement. The license fee is part of the purchase price of the Designated System.

Title

The original, and any copies of the Licensed Program, in whole or in part, which are made by Licensee, are the property of Licensor.

Copies

With each license, Licensee may make one (1) copy of the Licensed Program in object code form only for use by Licensee with the Designated System for backup or archive purposes. Licensee agrees to maintain records of each copy of the Licensed Program and the serial number of each computer system with which the Licensed Program is incorporated or used.

Copyright Protection

The Licensed Program is copyrighted by Licensor. Copies may be made only as permitted by this Agreement. Licensee agrees to reproduce and apply the copyright notice and proprietary notice of Licensor to all copies, in whole or in part, in any form, of Licensed Program made hereunder.

Warranty Exclusion

Any maintenance obligations of Licensor shall be subject to a separate maintenance or update agreement between and licensee.

LICENSOR MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED ON ANY LICENSED PROGRAMS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Patent and Copyright Indemnification

Licensor shall indemnify Licensee against liability for patent and copyright infringement upon the terms and conditions applicable to Licensee's purchase of the Designated System.

Termination

This agreement and any licenses granted hereunder may be terminated by Licensor upon written notice if Licensee fails to comply with any of the terms and conditions of this agreement. Upon termination of any license, Licensee will, at Licensor's option, either return the Licensed Program to Licensor, or destroy the original and all copies and parts thereof.

General

This agreement is not assignable. The rights under this Agreement or any license granted hereunder may not be assigned, sublicensed or otherwise transferred without the prior written consent of Licensor. Except as otherwise specified herein, this is the entire agreement between the parties relating to the subject matter hereof and may only be modified in writing signed by each party. This agreement is governed by the laws of the State of California.

For further information regarding this Cortex 1.0 User's Manual or other products, please contact:

Motion Analysis Corporation
3617 Westwind Boulevard
Santa Rosa, CA 95403 USA
tel: 707.579.6500
fax: 707.526.0629
info@motionanalysis.com

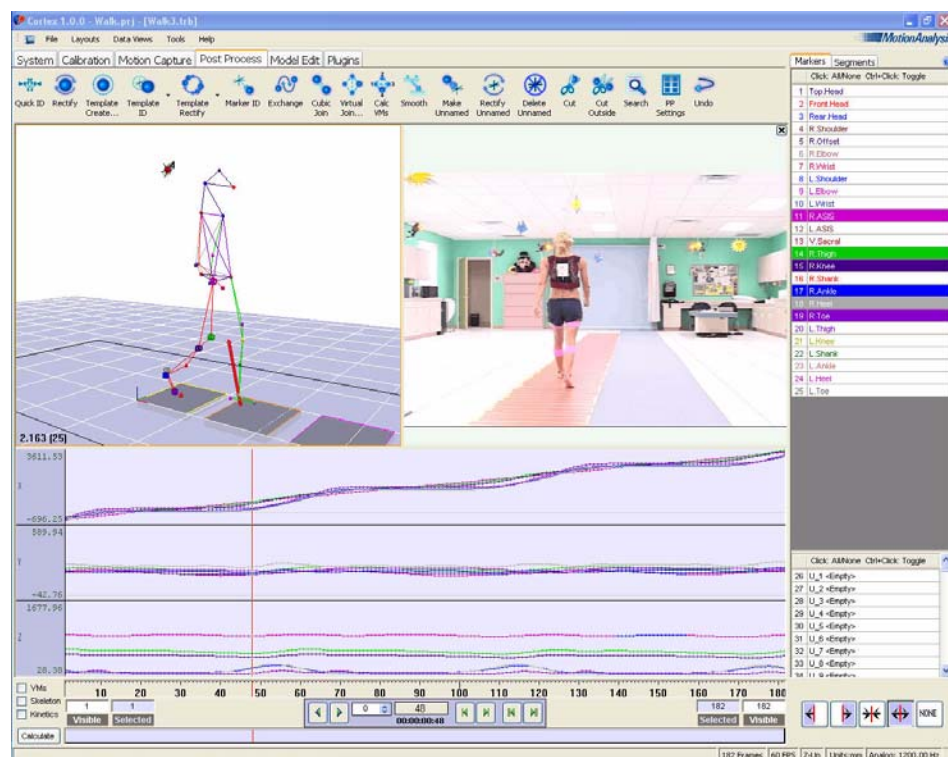
P/N 665-1200-010
www.motionanalysis.com
Copyright © 2008

Chapter 1 Introduction

Topic	Page
Overview	1-1
System Requirements for Cortex	1-3
Installing the Software and Licenses	1-4
Software Packages within Cortex	1-7
Software Packages Used with Cortex	1-12
For More Information	1-18

Overview

Figure 1-1. Cortex User Interface



This instructional User's Manual provides a complete description of the **Cortex** software and its capabilities, along with many step-by-step procedures critical to a successful motion capture project. Motion capture theory is separated from the body of this manual in the form of appendices so that the tutorial approach does not become cluttered.

Cortex is a complete package, capable of meeting the most demanding requirements of the motion capture industry. Output is generated in real-time making **Cortex** a suitable engine for a number of widely used 3D animation packages as well as custom applications created using the supplied Software Developers Kit (SDK). Being a real-time application, the results of a motion capture session can be viewed instantly while simultaneously saved in several file formats. In addition, you can graphically edit data with a complete suite of tools without resorting to other off-the-shelf software packages.

Cortex handles image data from systems comprised of up to 250 cameras. System setup and calibration is fast and simple with immediate feedback and a high degree of accuracy and precision. Motion capture sessions are managed using directory and file access tools and the motion data generated is of the highest quality. Post Processing data is accomplished graphically using intuitive controls integrated with mouse and keyboard functions for fast and easy editing. Model Edit features give you access to the properties of the current set of named markers, virtual markers, linkages, and skeletal segments.

Cortex combines three major functions into a single software package:

1. Calibration of your capture volume
2. Tracking and identifying marker locations in your calibrated 3D space
3. Post processing tools for tracking, editing, and preparing the data for other packages

The advanced calibration procedures calibrate the 3D volume with ease and accuracy.

Options for using the software include:

1. Color video capture-synchronized with the **EVaDV** software—either on your capture computer or on one or more auxiliary computers
2. Synchronized analog channels—capable of collecting from 32 to 192 channels of analog data at any frequency between 60 and 5000 Hz
3. Genlock to your studio camera with the Eagle and Hawk cameras
4. Calcium/Solver—for generating constant bone-length skeletons for high quality animation
5. OrthoTrak—clinical gait evaluation module
6. SIMM—Software for Interactive Musculoskeletal Modeling
7. KinTrak—research module for defining research projects with kinematic and kinetic data and a trial and subject database
8. Kinetics—a Real-Time kinetics and kinematics calculation engine
9. Motion Composer—used for producing, packaging, and presenting data reports

System Requirements for Cortex

Table 1-1. Required Minimum and Recommended Specifications

Required Minimum Specifications	Recommended Specifications
Dual 2.0 GHz CPU (up to 12 cameras)	Dual 3.2 GHz CPU, Dual Dual-Core
2 GByte RAM	2 to 4 GByte RAM
Windows XP Pro™ with .NET 2.0 or later operating system	Windows XP Pro™ with .NET 2.0+ or Windows Vista 32 Business™ or Windows Vista 64 Ultimate™
OpenGL video card with 128 MByte RAM capable of (1280x1024) resolution	OpenGL video card with 256 MByte RAM capable of (1280x1024) resolution
19-inch or larger monitor capable of (1280x1024) resolution	Dual 20-inch or larger monitors capable of (1280x1024) resolution
100 GByte hard drive (IDE or SCSI)	200 GByte hard drive (IDE or SCSI)
USB 2.0 Ports (x3)	USB 2.0 Ports (x4)
Internal CD-RW drive	Internal CD/DVD-RW
1 Gigabit Ethernet Network Interface Card (NIC), quantity dependent on the following options: <ul style="list-style-type: none"> Connecting to camera network Connecting to Internet Streaming to SDK or plugins 	2 Gigabit Ethernet Network Interface Card (NIC), quantity dependent on the following options: <ul style="list-style-type: none"> Connecting to camera network Connecting to Internet Streaming to SDK or plugins
104 key keyboard	104 key keyboard
Three-button mouse. The program requires a middle mouse button for zooming and selecting in several of the graphical panes.	Three-button mouse. The program requires a middle mouse button for zooming and selecting in several of the graphical panes.
IEEE 1394 (standard Firewire) for Reference Video option	IEEE 1394 (standard Firewire) for Reference Video option

Hardware

Cortex will perform best on a dual processor host computer with an OpenGL graphics card. This is used for all Motion Analysis camera types.

Using Raptor, Eagle, and Hawk Digital Cameras

The Raport, Eagle, and Hawk digital camera motion capture system includes a set of Eagle and Hawk digital cameras with ring lights, LAN/power cables, and an EagleHub system.

Using Falcon Cameras

Cortex does not support collecting data with analog camera systems (Falcon, Cohu, or Pulnix cameras using the Midas computer). However, you can use **Cortex** for post processing and analysis of motion capture data captured with these analog systems.

Middle Mouse Button

For **Cortex** operations, the middle button is key for zooming and translating through the 3D and XYZ Graphs display. You will need to verify that the middle mouse button is set to the middle button function.

Software

The **Cortex** program requires the host **Windows XP Professional™**, or **Windows Vista™** operating system.

Installing the Software and Licenses

To install **Cortex** for the first time, simply insert the installation CD-ROM into your computer and select the **Setup Cortex No Samples.exe** or **Setup Cortex With Samples.exe** file

Note: To run **Cortex**, you will need both a license file and a dongle from Motion Analysis Corporation. The license file you receive is keyed to your Motion Analysis dongle number printed on the dongle.

Figure 1-2. USB Port Dongles and Flash Drive



USB Port Dongles

Motion Analysis Flash Drive

Installation Using the Flash Drive

For new users, **Cortex** licensing is now provided and setup using Flash Drives. For installation, please follow these steps:

Note: Please make sure to remove all Dongles from your computer prior to running the **Cortex** setup file on the CD. Failure to do so may result in damage to your dongle.

1. Install the new version of **Cortex** from the CD.
2. Install the Sentinel Drivers after the **Cortex** installation is finished. Note that when installing the Sentinel Drivers, select the custom install and uncheck the option for "Sentinel Protection Server."
3. Insert the Dongle into the USB port
4. Insert the Motion Analysis USB Flash drive into a USB port. Double-click on the program: Install.Mac.License and follow the instructions. Press **Y** or **Enter** and the license will be installed onto your hard drive, in the **C:\Program Files\Motion Analysis** directory.
5. On the Task bar, left-click the Green arrow icon and select **Stop USB mass Storage Device**.
6. Unplug the USB Flash Drive and store it in a safe place.

If you need any further information, please consult the **readme.txt** file located on the Motion Analysis USB Flash Drive.

If You Already Have Software Installed

If you have Motion Analysis software already installed, you will need to add a new line to your Motion Analysis license file (provided by Motion Analysis Customer Support).

1. Launch **Notepad**, **Wordpad**, or your favorite ACSII text editor.
2. Navigate to **C:\Program Files\Motion Analysis**.
3. Open the **mac_lic.dat** file.
4. Add the new line beginning with **[Cortex 1.0]** from your new license to the bottom of your current license as shown below.

Figure 1-3. Sample Motion Analysis License File

```
Motion Analysis License File
Customer: MAC Customer
Platform: NT
SystemID: 19c
Created: 9/15/20xx 1:42:26 PM
Sales Order#: 08-xxx
Entered By: Support
[Cortex 1.0] aed50167 873b2d56
[Analog Input] b9806c31 d1567841
[OrthoTrak] b2df5e69 8964274a
[Animation Plugins] b1a50160 805b5c49
[Director/Sequencer] e1a04e65 85745819
[RT2 Animation Plugins] e3f05340 a069081b
[Analog Input] b9806c31 d1567841
[Calcium 4] e7ed5923 c363151f
[Skeleton Builder 4] a3f44279 99780c5b
[Reference Video 3.0] eb92592f cf636a13
[Talon Streaming 4] ecb36136 d65b4b14
[Talon Viewer 4] 86fb0714 f43d037e
[Motion Composer] c7f00e25 c534083f
This license has no expiration.
```

If You Install a Dongle

Computers with a new dongle installed need to load the dongle drivers so that the application will detect the dongle.

You can choose to install the dongle drivers when initially installing the **Cortex** software or you can run the drivers independently by running the program in the **Sentinel Drivers** directory under the **Cortex** folder.

Operating Systems in Different Languages

If the operating system you are using on your tracking or post processing computer is a non-English version, some characters may not be recognized and you may experience installation problems.

If you are experiencing this, you will need to go into the computer and set it to allow for English Unicode characters. In **Windows XP**, you can do this by going to **Start > Control Panel > Regional and Language Settings**. This brings up a window that has 3 tabs, and the second one is the Language Tab. Under the Text Services and Input Languages tab, you need to click on the **Details** button and add the setting for English (United States). This will add the necessary text characters to the computer.

Alternatively, you can install an English language OS on your computer.

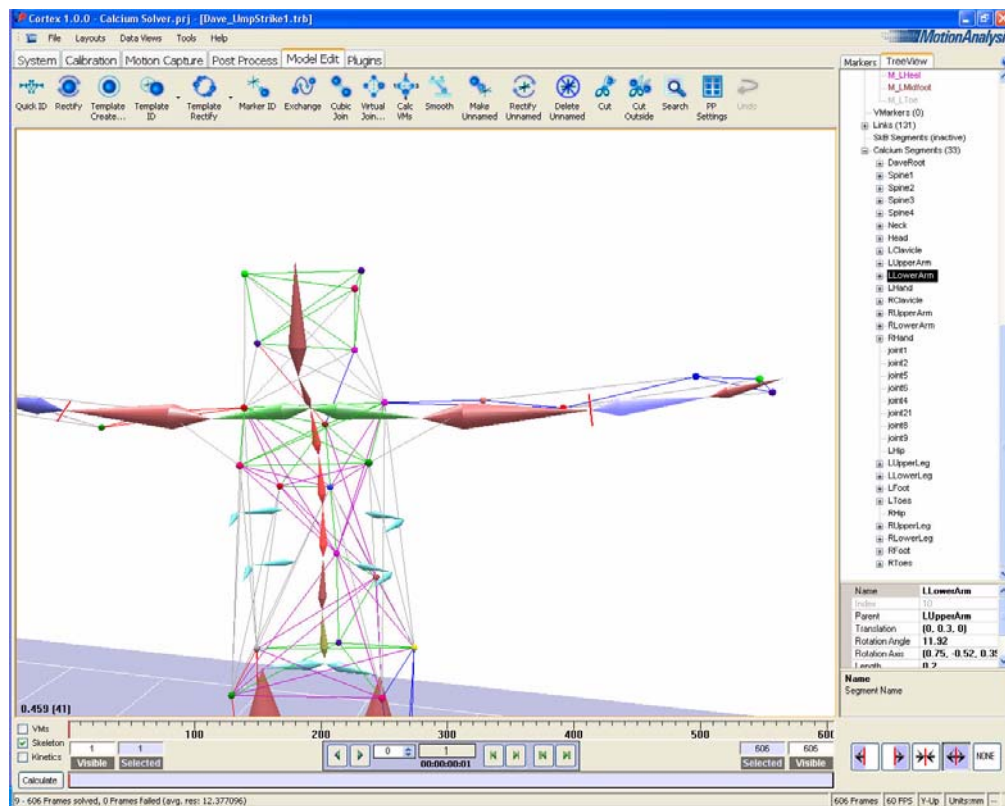
Software Packages within Cortex

The following are software products offered by Motion Analysis that are integrated within the **Cortex** user interface. These files will require a license file and a .dll file to use.

Calcium

Calcium is the graphical user interface to the Solver engine. Solver is a powerful numerical tool for calculating skeleton motion from marker data. The Calcium interface in **Cortex** is what allows you to correlate the positions of a marker pose to the initial pose of a skeleton. The skeleton is usually created in an outside animation package, such as Maya, 3D Studio Max or Kaydara and then exported to an HTR file by a Motion Analysis file IO plugin for that package.

Figure 1-4. Calcium Interface in Cortex



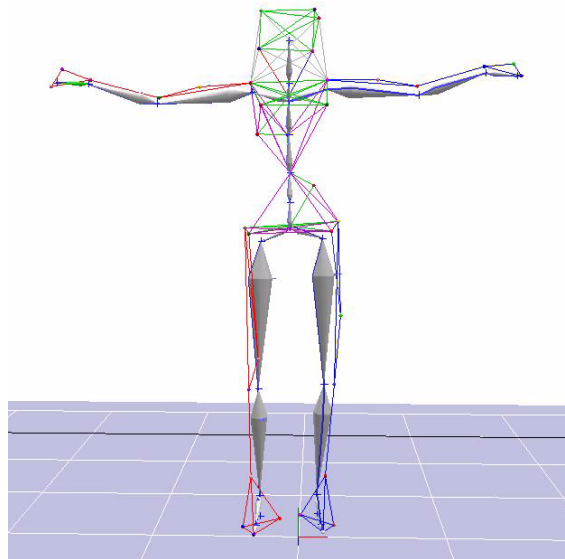
Note: Solver Interface (Si) is the same software as Calcium, only that it's interface is separate from **Cortex**.

Skeleton Builder (SkB)

A skeleton, in animation terms, is a hierarchically connected set of bones with translation and rotation data. Each bone has a parent and potentially any number of children. One special bone has no parent and is usually referred to as the "root" of the skeleton. Skeleton Builder, as the name implies, is a tool that allows you to construct a skeleton by creating bones and arranging them in a hierarchy. Each bone is defined by the motion of three markers used to construct its rotation data. The Skeleton Builder interface is incorporated with the **Cortex** interface. Skeleton Builder is typically used in the Movement Analysis (Biomechanics) applications.

Skeleton Builder bone definitions are stored in the **Cortex** project file. Any time you wish to save the definitions you have created simply save out a project file. Various project files are stored in the sample directory which contain the example skeleton at various stages of construction.

Figure 1-5. SkB Skeleton



Motion Composer

Motion Composer is a suite of tools for collating, integrating, and presenting interactive motion capture data. Motion Composer is a collection of authoring tools, data structures, and visualization panes. These are integrated into **Cortex** to help achieve a seamless workflow for the user to package and present a motion capture session. Some of the key features to be found in Motion Composer are described in the following sections.

Integrated Authoring

Motion Composer is designed to be used within **Cortex**. This integration allows new and existing users a seamless pathway from data collection to collation and presentation with a minimal learning curve. For current users, this integration leverages their existing knowledge of **Cortex**. Presentation output can be as simple as redirecting an **Cortex** project to presentation format.

Interactive Player

Motion Composer ships with Motion View, a freely distributable interactive player that enables customers to distribute their presentations quickly and easily. When it's time to send research data to a colleague or take it on the road, authors can simply pack-and-go, turning their presentation into a single packaged ZIP file for quick burning to CD or emailing to a friend. With the interactive player inside, presentations are ready for launch on any Windows XP/Vista operating system.

Rich Media Support

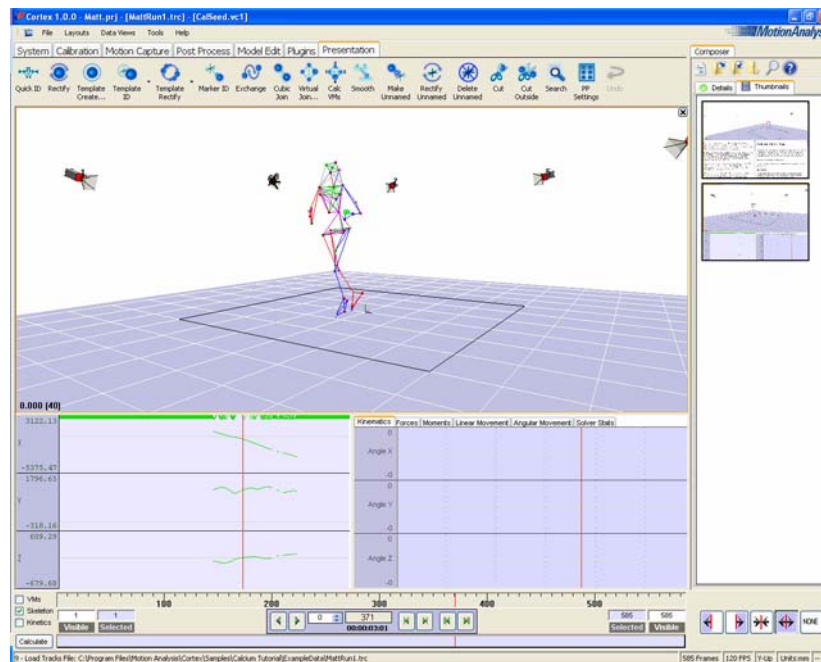
Motion Composer supports the import of a wide variety of information formats, allowing users to import not just motion capture data but data from third party applications, such as EMG analysis graphs, additional color video footage, and still images. It also allows users to import user generated data such as clinical notes, Microsoft Word documents, and embedded HTML link. Below is list of the file formats Motion Composer supports:

- Text (TXT, RTF, HTML, XLS)
- Color Video (MPG, AVI)
- Images (JPG, GIF, BMP)
- Project files (PRJ)
- Tracks (TRC/TRB)
- Analog data (ANB)
- User defined data types/views

Presentation Tools

The Motion Composer authoring interface provides a simple hierarchical interface to easily manage a user's disparate files, offering views of referenced files, data structures (e.g. joint angles or segment angles), or relationship structures (e.g. Subject/Condition/Trial/Cycle). The Motion Composer authoring interface also makes it simple to create presentations. As data is added, the author simply creates a view of the data they wish to convey and stores this view as a slide. Presentation viewing becomes as simple as a slideshow.

Figure 1-6. Motion Composer Interface



Digital Video Option (EVaDV Software)

The color Digital Video option allows you to record a time-matched Reference Video along with your motion capture trial on a separate computer. With this option, you will record a time-matched color video AVI file with the same trial name in your motion capture folder. A separate computer is used in order to not burden your **Cortex** Host computer, which is an issue if your computer is too slow for the number of markers being tracked. For single person captures, you may connect the DV Camera directly to the **Cortex** Host computer. In this case, the **EVaDV** software is not needed as it is built into the **Cortex** software. You can run **EVaDV** on one or more computers and then capture multiple AVI files (multiple views). You may experience a small delay in frames from the **Cortex** software and the **EVaDV** software when capturing. This can be easily reconciled using the Adjust Frame Offset function (found in **Tools > Settings > Playback**), which allows for time-matching data streams.

Note: It is recommended that you start your digital video collection from pause mode (not in run mode) to minimize the potential for frame offset between the AVI and TRB files.

Note: It is also recommended that you set the motion capture frame rate to a multiple of the DV frame rate:

- NTSC: 29.97 Hz to 30 Hz
- PAL: 25 Hz

Software Packages Used with Cortex

The following are software products offered by Motion Analysis that are used in conjunction with **Cortex**. These files will require a license file and a separate installation package.

Animation Plugins

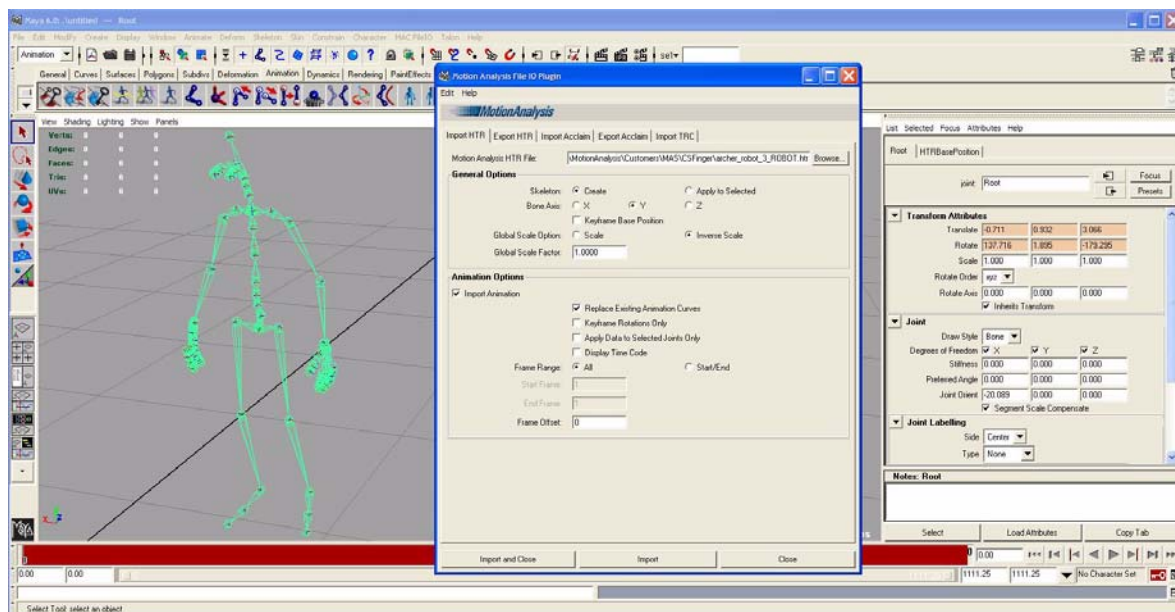
The MAC Animation Plugins, also known as the File IO Plugins, are used to read and write Motion Analysis motion capture files (TRC or TRB) into all of the major animation software packages including:

- Maya
- 3DS MAX
- Softimage XSI
- Lightwave
- Alias Motion Builder.

The Motion Analysis HTR file format is used for skeleton motion capture data. The Animation Plugins can take an existing character skeleton and export it to an HTR file (usually for use in Calcium) and it can take an HTR file created by Calcium and apply it to the character in the animation software.

The Motion Analysis TRC file format is used for tracked (markers) motion capture data. This data is generally only imported into the animation software. It is used for bringing in full body marker data, face data and prop data.

Figure 1-7. Animation Plugins Interface

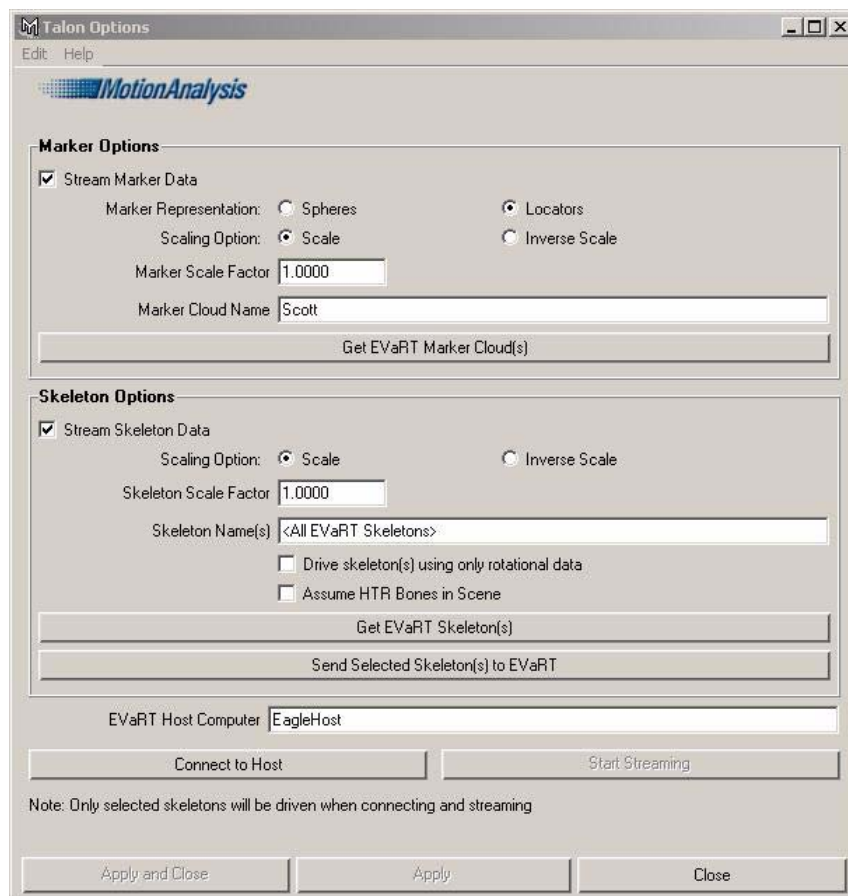


Talon Plugins

The MAC Talon Plugins, also known as the Streaming Plugins, are used to stream data from a live, realtime connection to **Cortex** into an animation software package. Both skeleton and marker data can be streamed. This function is available for Maya, Mocap, and 3DSMAX. Additional animation packages are currently under development.

This same interface is used by outside developers to stream motion capture data into their own custom environments. This programming interface is called the SDK and is available upon request.

Figure 1-8. Talon Plugins Maya Interface



OrthoTrak

OrthoTrak is a completely integrated full-body gait analysis package designed for use in clinical and research studies of human locomotion. The system provides state of the art software designed to be used by clinicians in orthopaedics, neurology, and physical therapy, or for any person interested in assessing locomotor abilities of humans.

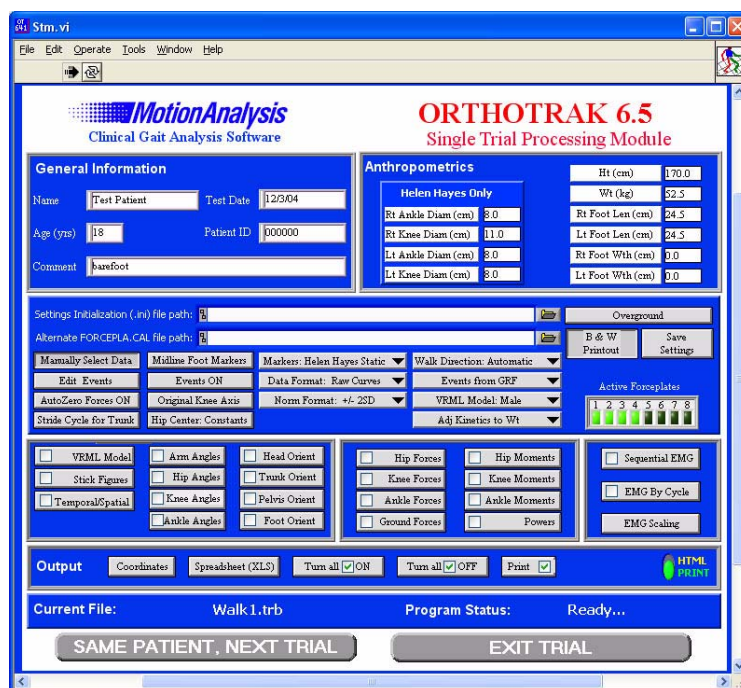
The system provides:

- quantification of 3D body, segmental, and joint motions
- analysis of the forces occurring in locomotion
- records of neuromuscular function through electromyography

Intended Use

OrthoTrak is designed primarily for analyzing a walking motion over level ground but can also be used for walking on treadmills, up and down stairs, or other activities. Gait data are presented in graphs which describe the kinematic (i.e. joint angles), kinetics (i.e. moments), and muscle activity (EMG). Data can be exported as industry-standard XLS data sets which can easily be imported into **Microsoft Excel™** or other graphics and analysis packages.

Figure 1-9. OrthoTrak Interface



KinTrak

KinTrak is a movement analysis system that enables you to import and analyze three-dimensional kinetic (i.e. moments), kinematic (i.e. joint angles), and analog channel (e.g. EMG) data from a biomechanical perspective.

Existing biomechanical software tends to be geared toward solving specific problems, and is not usually adaptable in new situations. This has resulted in programs that are too restrictive in data control, collection and analysis. **KinTrak** is designed to overcome these problems by allowing users to tailor the program to suit their specifications and the requirements of the project or study they are undertaking.

KinTrak is intended to be used by researchers and clinicians in the fields of biomechanics and human movement. In addition, coaches and assistants of sports teams may find it useful for their purposes. Possible uses of **KinTrak** include studies for the purpose of prevention of athletic injuries, analysis of gait characteristics, assessment of athletic footwear, and development of prosthetic appliances.

Director/ Sequencer

Director/Sequencer allows the animator to perform *non-linear* editing of motion data on screen. Just as a sound engineer uses a multi-track digital workstation to edit and combine many tracks of sound, the **Director/Sequencer** gives the animator the ability to do the same with motion captured data.

The animator starts with a collection of files containing motion data for an object or a hierarchical skeleton. Each file that is loaded is displayed as a move in **Director/Sequencer**. Each move is represented as a time line in an **Editor** area and a 3D character in a **Viewer** area.

By simply pointing and clicking with the mouse, the animator can slide a move backward or forward in time in the Editor area. In the Viewer area, both the position and orientation of each figure can be modified. Two moves can be joined together so that there exists a user defined region where one is blended into the other. Finally, any part of a figure's motion can be hand edited to give the animator complete control of the finished animation.

Each move is stored as a Hierarchal Translations and Rotations (*.htr or HTR) file which contains motion data for a hierarchical skeleton. After all moves have been choreographed, they are once again saved as HTR files which can then be used as input to one of the popular animation software packages to generate the final animated scene.

SIMM

SIMM (Software for Interactive Musculoskeletal Modeling) is a software system that enables you to create and analyze graphics-based models of the musculoskeletal system. In **SIMM**, a musculoskeletal model consists of a set of bones that are connected by joints. Muscle-tendon actuators and ligaments span the joints. The muscles and ligaments develop force, thus generating moments about the joints.

SIMM allows you to analyze and test a musculoskeletal model by calculating the moment arms and lengths of the muscles and ligaments. Given muscle activations, the forces and joint moments (muscle force multiplied by moment arm) that each muscle generates can be computed for any body position. By manipulating a model on the computer graphics system, you can quickly explore the effects of changing musculoskeletal geometry and other model parameters.

Since the software can be used to study many different musculoskeletal structures, it can enhance the productivity of investigators working on diverse problems in biomechanics. **SIMM** provides a framework that organizes the parameters of a model and allows people to work together on a modeling project. The moving, three-dimensional images of anatomical structures that you can create are extremely valuable when developing a model and when communicating the results of an analysis.

Applications

SIMM has a wide variety of applications. A few examples include the following.

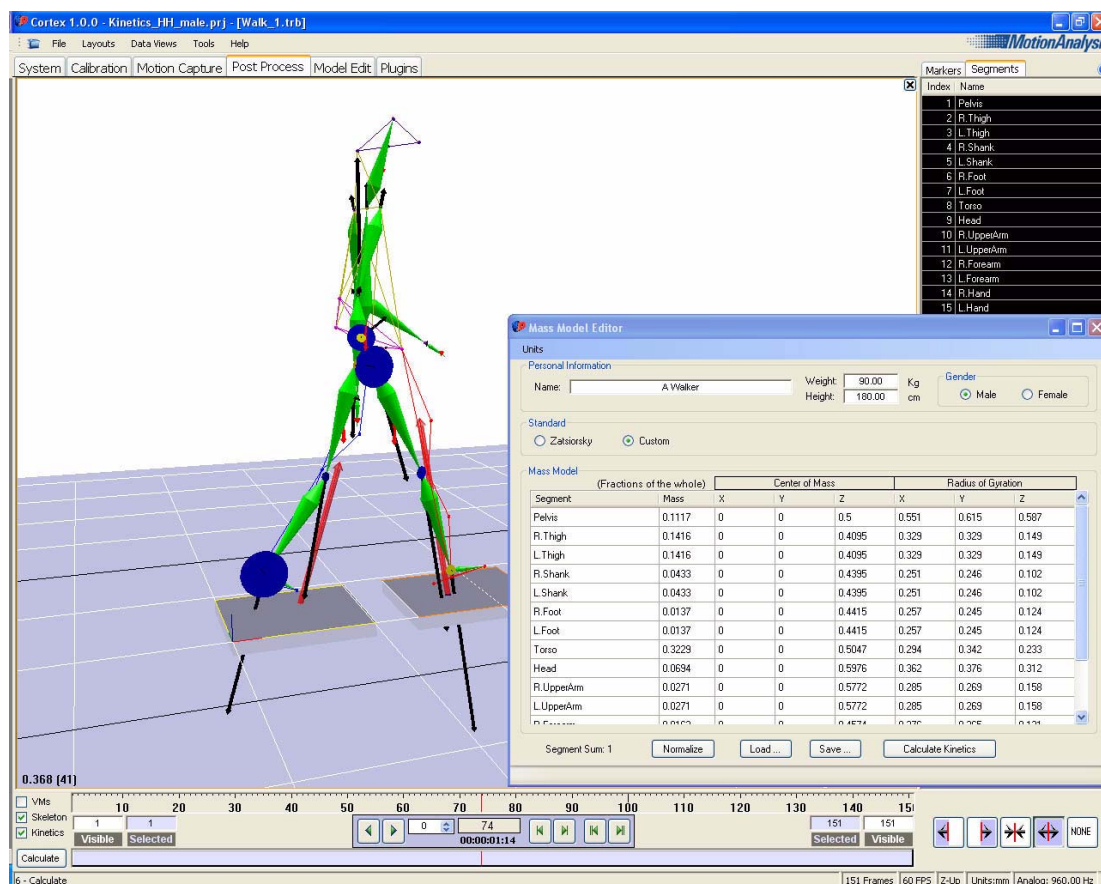
- Biomechanics researchers are using **SIMM** to create models of the human elbow, wrist, jaw, and other anatomical structures. These models can be altered according to particular surgical procedures to study how the surgical alterations affect muscle function. **SIMM** can also be used to analyze and display the mechanics of injuries.
- Neuroscientists are using **SIMM** to study how the central nervous system controls movement. For example, muscle activation patterns determined from electromyographic recordings can be used to estimate muscle forces and joint moments generated during a task. The computed joint moments can then be compared to experimentally recorded moments.
- Medical students and residents can use models created with **SIMM** to study musculoskeletal anatomy and function. In addition to visualizing anatomical structures, students gain an appreciation for the interplay of muscle architecture and joint geometry.
- Kinesiologists who record and analyze the motion of persons with movement disabilities can use **SIMM** to create three-dimensional animations of a person's movement. Movements, such as walking, can be quantitatively compared to normal movement to gain insight into the causes of movement deformities. Motion can also be analyzed in the context of optimizing athletic performance.
- Human factors engineers who need to account for muscle strengths when designing products or work stations can use **SIMM** to study how posture effects muscle strength. Limits on joint ranges of motion can also be taken into account.
- Biologists interested in animal movement can create models to quantify limb function. Investigating movement strategies in other species can provide insights needed to design machines that move.

- Computer scientists who develop models of the human body for virtual environments can use **SIMM** to create the models and compare them with biomechanical data for verification.
- Animators can use **SIMM** to develop realistic representations of human and animal movements. World objects can be added to provide a context for the animation.

Kinetics

Kinetics is a full-body, three-dimensional engine capable of calculating kinematic and kinetic information from models that are created and saved in the project file(s). The kinetic calculations can be done on either of the two skeleton types available in Cortex: Skeleton Builder and Calcium. The Skeleton Builder models are generally simpler to use as they automatically scale the bone lengths to the subject's actual bones. The Calcium based models use an entirely different computation method of Global Optimization. With either skeleton type used, the end results are similar.

Figure 1-10. Cortex Kinetics Interface



For More Information

Please contact Motion Analysis Customer Support with any questions, problems, or feedback about the Motion Analysis **Cortex** software. We can be reached at:

Motion Analysis Corporation
3617 Westwind Blvd.
Santa Rosa, CA, USA 95403

Phone: 707-579-6500
Fax: 707-526-0629

<http://www.motionanalysis.com>

For technical support and licensing information, please contact:

support@motionanalysis.com

For information about sales, please contact:

info@motionanalysis.com

Acknowledgments

Logo and cover artwork—Kristopher Evans, Motion Analysis Studios.

Also, many thanks to the users and Beta testers listed in **Help > About Cortex**.

Quick-Start Tutorial for Movement Analysis Applications

Topic	Page
Overview	2-1
Starting Cortex	2-1
Project Initialization	2-2
System Calibration	2-3
Setup Analog	2-6
Marker Placement	2-8
Data Capture	2-10

Overview

This chapter provides a quick reference to begin using your motion capture system for Movement Analysis applications, and is intended for the more advanced motion capture system user. For Animation Production applications, refer to Chapter 3, Quick-Start Tutorial for Animation Production Applications.

Note: This Quick-start Guide uses a Helen Hayes marker set and starts with a project file that is located in the **C:\Program Files\Motion Analysis\Cortex\Samples\Helen Hayes Markers** folder. The basic methodology outlined here can be generalized to other marker sets. Please be aware, that this data set is more complex since it utilizes two different marker sets (Static and Dynamic), and captures three different sets of data (Static, Dynamic Template, and Dynamic movements).

Starting Cortex

1. Turn on the Host computer and login.
2. Turn on the Ethernet switch and CP-8 Power Hub. The cameras will automatically power up.
3. Turn on the Forceplate and EMG Amplifiers, if applicable. Make sure to zero the force plate(s) (see the manufacturer manual for more information).
4. Launch the **Cortex** software by double-clicking the icon located on your computer's desktop.

Project Initialization

1. Load a previous project (**File > Load Project**) that has a Helen Hayes static marker set (e.g. **Static.prj**).
 - Project files contain information about calibration, thresholds, masks, tracking parameters, marker sets, and templates.
 - By loading a previous project that contains all of this information, you will not have to re-enter it all each time you start a new capture session. You will only need to update the calibration, if necessary.
2. Immediately save the project file in a new folder (**File > Save Project As...**).
 - Create a new folder for the subject and save your project there. This directory now becomes the default **Cortex** directory.

Note: Make sure that you do not write over previous projects. Separate projects are needed in order to run trials for that particular day. If calibration VC files are written over, then recreating the calibration parameters in Post Process mode will not be possible.

3. Choose **Frame Rate** and set the shutter speed, brightness, and threshold.
4. Press **Connect to Cameras**.
 - The first time you do this step, a message indicating that “X number of cameras were found, existing project has 2. Do you want to modify project?” may appear. See [Figure 2-1](#). Press the **OK** button.

Figure 2-1. Connect to Cameras Status Pop-Up (Example)



System Calibration

Note: System calibration should be done at a camera speed of 60 Hz, regardless of the data capture speed.

Start the Calibration Process Using the (4-Point) Calibration L-Frame

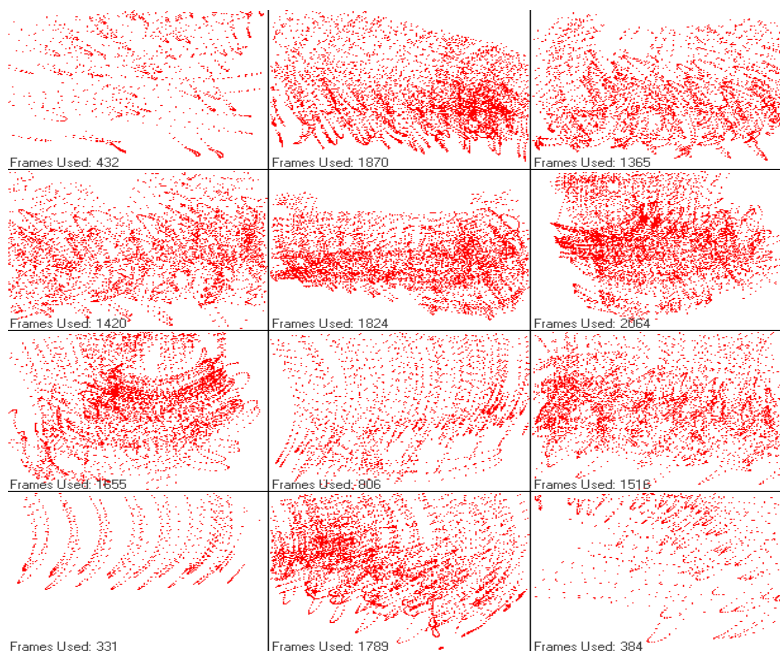
1. Place the calibration L-frame device or four markers (L-shaped) on the floor or on the forceplate.
 - The calibration L-frame markers are set under **Tools > Settings > Calibration** tab. These markers have been placed in a particular orientation and precise distances apart in order to tell the software the origin and coordinate (XYZ) system of the lab/room.
2. Under the **Calibration > Calibrate** panel, activate the Camera Aiming check-box. Note that this will erase any previous calibration files.
3. Press the **Run** button.
 - All the number buttons on the bottom of the **Cortex** interface should turn yellow if all of them can see the L-frame.
4. Select **Layouts > 2 Panes: Top/Bottom**.
 - We want the 3-D Display window and 2-D Display window showing. These can be set by left-clicking in the window to make it active and then select **Data Views > 3D View** or **2D View**.
5. Check the 2-D views for each camera.
 - Generally, there should only be four markers in each camera view. If there are less, you may need to adjust the view of the camera or you can also adjust the threshold to see more markers. If you have more, you can mask out extraneous data points. To mask, press **Pause**. While in one of the 2-D views, press the middle mouse button and hold it, then drag a square over the bad data to mask.
 - To cover all areas in a large capture volume, you may need to aim some of the cameras where they will not see the Calibration L-frame. The **Cortex** software will then calibrate these cameras during the wand calibration procedure.
6. Check the 3-D display and camera locations.
 - If it is not already set, right-click and select **Show Cameras**. All of the cameras should be in the correct place.
7. Optimize the camera positions and their orientation.
 - New camera positioning should be done at this point if needed.
 - Right-click in the 3-D view and select **Show Camera Field-of-View**. You will probably have to change the length of the field of view to more than the default value of 4000 (4 meters). This adjustment slider is found in the **Tools > Settings > 3D Display** tab. Try 9000 (9 meters).
 - Turn the capture volume on by right-clicking your mouse in the 3-D view and selecting **Show Volume**. This volume is a visual aid helpful in this process of aiming the cameras properly. The volume dimensions are entered under **Tools > Settings > Calibration > Capture Volume** tab in the window.
 - Your camera field of view should cover the desired volume. Try and align edges of the volume box with edges of the camera field of view. This may require that one person moves the camera on 3 axes, while another person directs the movements.
 - Verify or set the focal length of the lenses under **Tools > Settings > Calibration > Lenses/Orientation**.

8. Once the camera position is optimized, press the **Collect and Calibrate** button in the Calibration with Square field. If sound is enabled and you have speakers turned on, you will hear a sound.

Continue the Calibration Process Using the Calibration Wand

1. Remove the L-frame from the capture area. It will need to be completely out of view from all cameras.
2. Set the wand length in the **Calibration with Wand** field.
 - Make sure the wand length is set at 200mm or 500mm depending on the wand used.
3. Set the capture duration.
 - The wand capture duration should be around 60 seconds, or long enough to cover the capture volume. During the 60 seconds, 1/3 of the time should be spent waving the wand parallel to each axis: x, y, and z.
4. Press the **Collect and Calibrate** button in the Calibration with Wand field.
5. Begin waving the wand to cover the capture volume as much as possible.
 - The object of this exercise is to cover the entire capture volume by waving the wand both horizontally and vertically through the cameras field of view. If you look at the 2-D fields of view, you should have only a small amount of white space for each camera. The better the coverage, the better the calibration.

Figure 2-2. Proper Wand Calibration Coverage



6. When finished, and the Wand Processing Status window appears, you can select the **Run Again** button. This will recalculate the calibration with continued emphasis on the wand data and will refine all the camera parameters. This step can be repeated several times until the calibration data (residuals) changes very little or becomes worse.

Figure 2-3. Wand Processing Status Window

Camera	56	57	58	59	60	61	62	63	64	65	66
U-Res.	29.098	35.391	38.107	75.929	7026.363	133.974	288.875	53.139	28517.930	160.277	-
V-Res.	19.547	21.659	49.253	53.909	4739.449	70.087	3795.858	50.210	7148.751	152.024	-
F. Length	18.000	18.000	30.871	30.253	18.000	32.967	18.000	31.759	18.000	36.000	18.000

3D Residuals
Avg: 165.262
Dev: 53.1826

Wand Length
Avg: 521.33
Dev: 74.47

Buttons: Run Again, Extend Seed, STOP, Accept, Reject

7. Check the calculated Focal Lengths.
 - The Focal Length for each camera is calculated and should be close to the value that is set on the lenses.
8. Check the 3D Residual Values:
 - The 3D Residual values should be small. The Standard Deviation should be approximately half of the 3D residual. Press the **Run Again** button until the values in the calibration processing window stop changing significantly or begin getting larger.
9. When everything looks good and you are ready, press **Accept**. If the calibration still does not meet the desired values, you can press the **Reject** button. You may have to do one part or all of calibration again.

Note: It is possible to calibrate with previously collected files.

10. Save the project (**File > Save Project**).
 - When you press **Accept** in the step above, you will get two messages stating “Calibration has been saved”. This message indicates that the project is saved to a system folder. You need to select **File > Save Project** in this step, since the system folder will be overwritten each time a calibration is done.

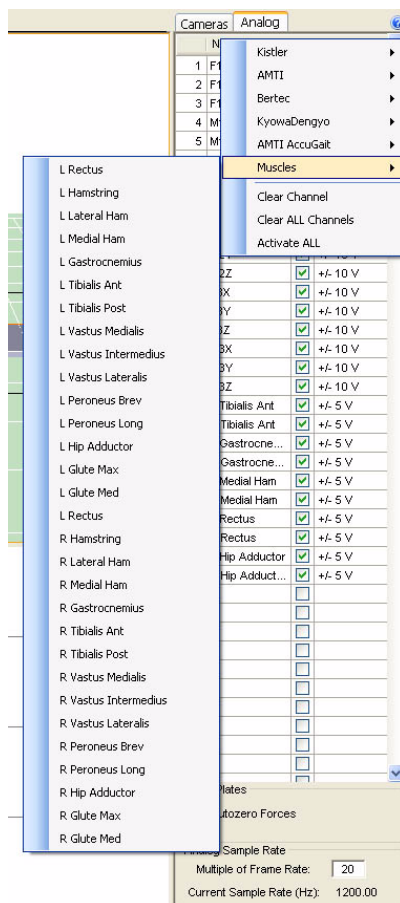
Possible Problems with Calibrations: How to Solve

- Wrong placement or measurements of the calibration L-frame. Verify all measurements and x, y, and z axes that are set.
- Check the brightness of the cameras and the use/non-use of masks. Remember to limit the use of masks and make them as small as possible if they are in line from the camera through the intended capture volume. If any markers go through a masked area, the data will be ignored.
- Too many extra marker images are possible causes for a bad calibration. Watch out for anything reflective such as extra markers, reflective material on shoes, shiny floors, debris in carpeting, and sunlight coming in through windows.
- If calibration problems persist, contact support@motionanalysis.com

Setup Analog

1. In the **System > Analog** panel, right-click on the Name column. Scroll down to **Channel type names** and select the type of Force Plate that you are using or if you are looking at lower body muscles, select **Muscles**.

Figure 2-4. System > Analog panel—Channel Type Names



2. If you are collecting forceplate data, then select one of the forceplate manufacturers and then select one of the FP1, FP2, FPx... choices available. These correspond to the number of forceplates you have available to use. This will automatically set 6 or 8 channel names (forceplate dependant) with default voltage settings for each forceplate selected.

Note: Do not enter the name of the forceplate manually. Use the predetermined settings from the menu.

3. For muscles, it will only bring up the muscle selected. The user may also specify their own name for an analog channel (e.g. upper body muscles).
4. The Range setting also has a drop down menu of varying excitation voltages. These setting should match your hardware (forceplates, EMG system, or other analog devices.)
5. Set your sampling rate at some value greater than the frame rate at which you are capturing video data.
6. To activate the channel name, simply click in the **On** column corresponding to the analog channel. A check mark will appear when active. Note that data will not be collected if no channel names are activated.

Marker Placement

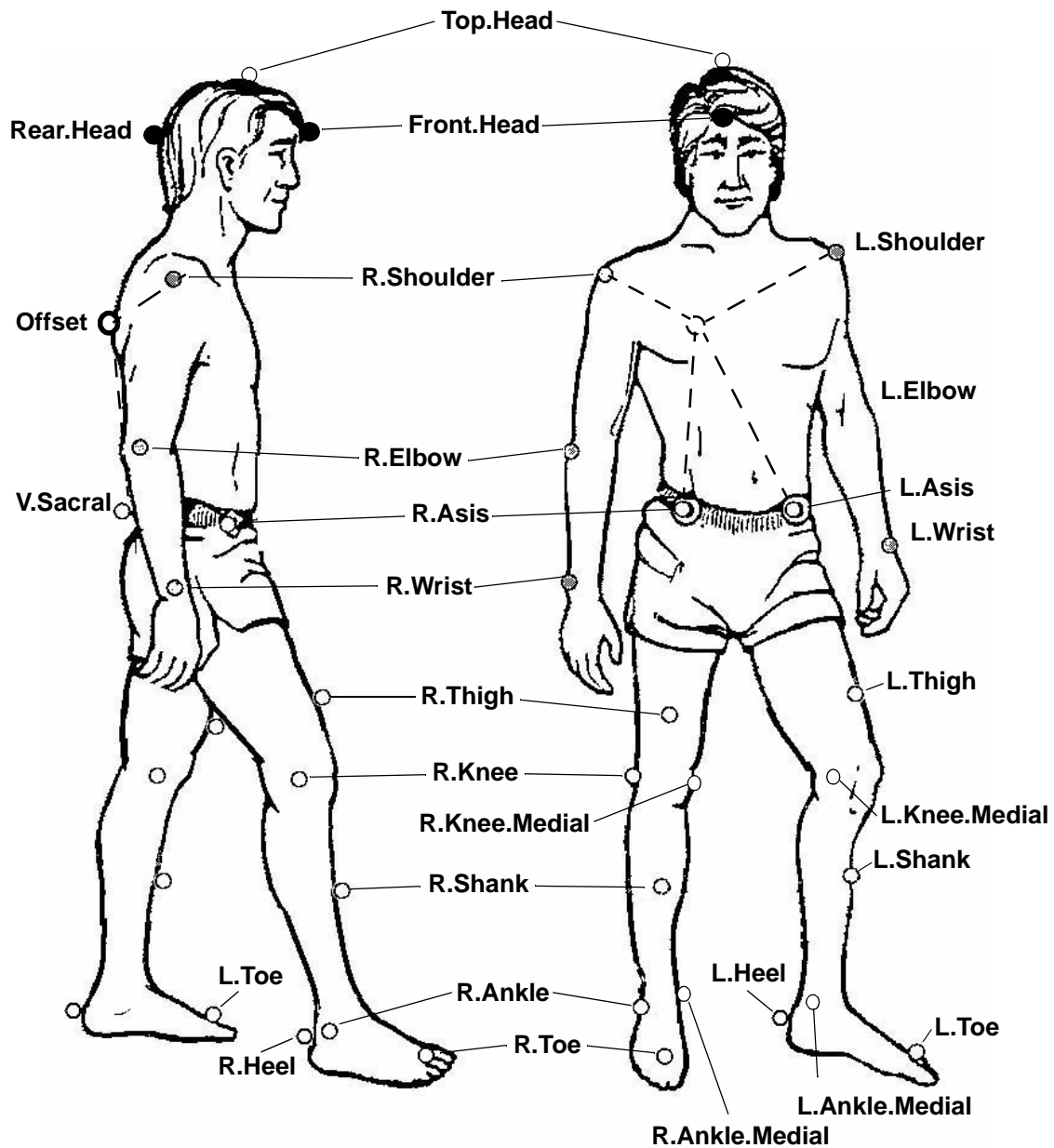
Note: For the purpose of this illustration, this example uses the Helen Hayes (both the static and dynamic) marker sets. The theory may be extended to different marker sets.

1. Attach the reflective markers that are listed in [Figure 2-5](#) and shown in [Figure 2-6](#). Placement on bony points is ideal if available. Consult an anatomy book as reference for palpating these points.

Figure 2-5. Helen Hayes Marker Set List

	Marker Names
1	Top.Head
2	Front.Head
3	Rear.Head
4	R.Shoulder
5	Offset
6	L.Shoulder
7	R.Elbow
8	R.Wrist
9	L.Elbow
10	L.Wrist
11	R.ASIS
12	V.Sacral
13	L.ASIS
14	R.Thigh
15	R.Knee
16	R.Shank
17	R.Ankle
18	R.Heel
19	R.Toe
20	L.Thigh
21	L.Knee
22	L.Shank
23	L.Ankle
24	L.Heel
25	L.Toe
26	R.Knee.Medial
27	R.Ankle.Medial
28	L.Knee.Medial
29	L.Ankle.Medial
*	

Figure 2-6. Helen Hayes Marker Set Placement



Data Capture

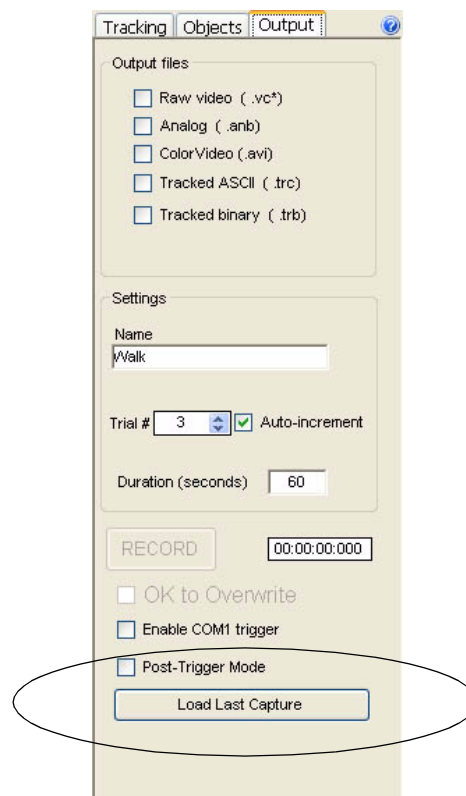
Capture a Static Trial

1. Go to the **Motion Capture > Output** panel and activate the Raw Video (.vc), Tracked Binary (.trb), and Analog Binary (.anb) check-boxes.
2. Type in a filename (e.g. Static). Do not use a number at the end of the filename. The trial number gets appended to the filename. If you need to have a number in the filename, make sure you follow it with an underscore (Static1_) otherwise your first trial will be interpreted by the software as trial 11 not 1.
3. Set the duration to be 1 second. Have the patient stand in the center of capture volume with arms raised parallel to the floor, thumbs facing forward.
4. Press the **Record** button.
5. This will produce a **Static1.trb** file and an analog and raw video file of the same name.

Identify a Static File

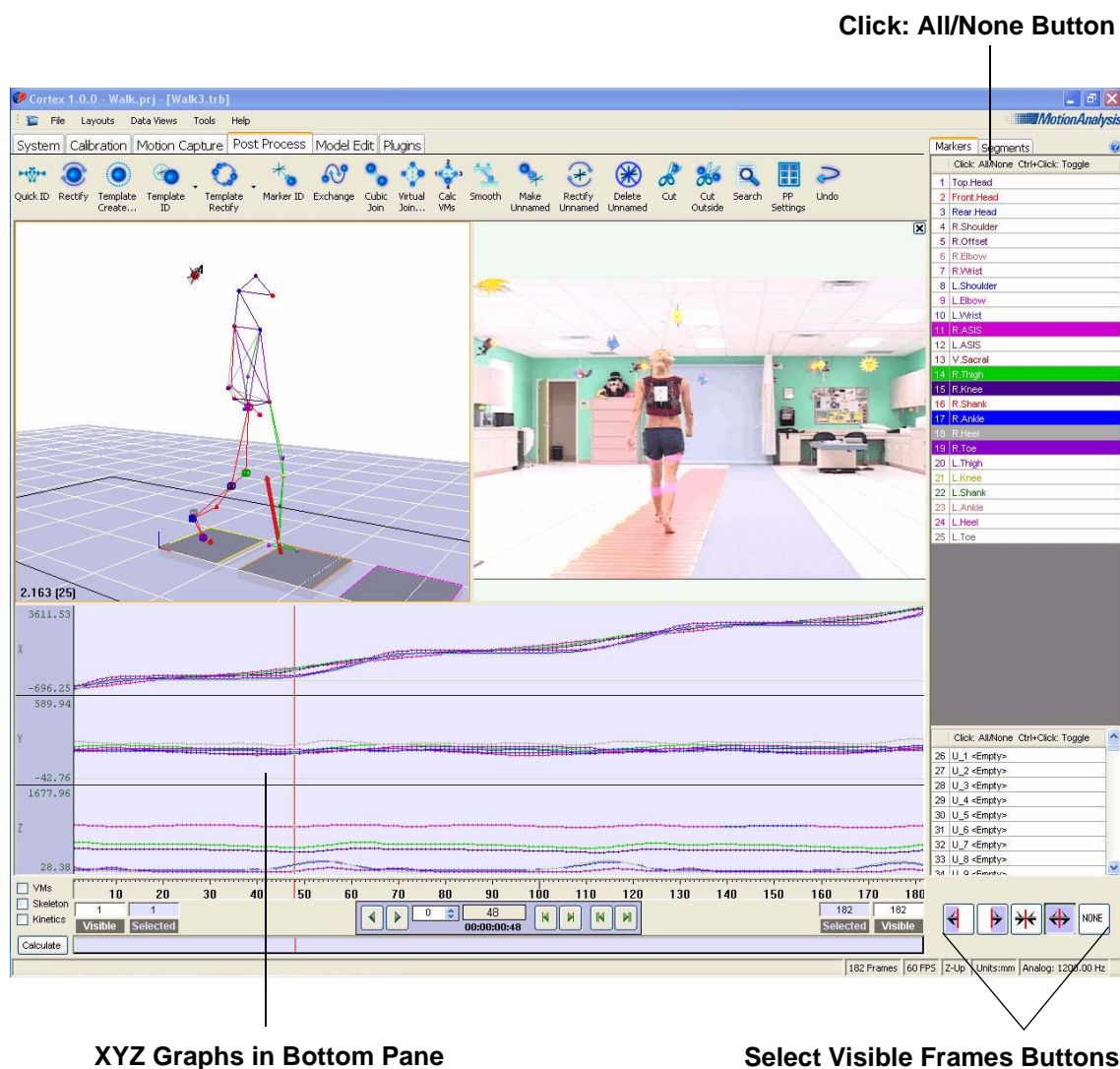
1. Load **Static1.trb**.
 - Do this either by pressing the **Load Last Capture** button on the **Motion Capture > Output** panel or go to **File > Load Tracks File**, then select and load it.

Figure 2-7. Motion Capture > Output Panel—Load Last Capture Button



2. Loading a file will automatically bring you under the Post Process interface. Select the **Identify** panel.
3. Press **Quick ID**. The Identifying window appears. Activate the **Rectify** check-box.
4. Identify each marker with the correct name.
5. Play the trial to make sure it is identified throughout the entire trial. If not, go to the frame you used for identification (usually frame 1) and press **Select Visible Frames** (shown in Figure 2-8) located in bottom right of screen and then press **Rectify**. Check again by playing the trial.

Figure 2-8. Post Process > MarkerSets Panel



6. Make sure there are no unnamed markers or gaps in the data. If so, activate the XYZ Graphs or in the bottom pane.

- Turn all markers on with the **Click: All/None** button, as shown in [Figure 2-8](#).
 - Press the **Select Visible Frames** button again, then press the **Delete Unnamed** button, which is located on the **Post Process** dashboard.
 - Finally, press the **Join Cubic** button.
7. Save this as a TRB file (**File > Save Tracks**).

Load a Walking (Dynamic) Marker Set

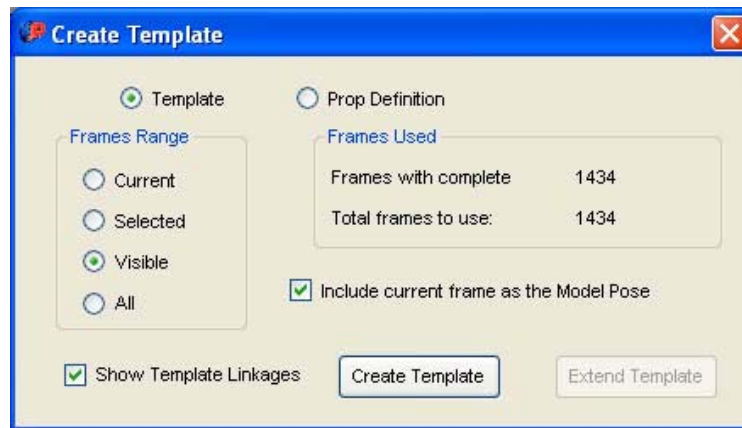
1. Select **File > Load MarkerSet**.
 - Load the **Walk.prj** file that contains the Helen Hayes Dynamic marker set. This will be the same marker set minus the medial knee and ankle markers.
2. Save the project (**File > Save As Project**) with your new name (i.e. Walk.prj), which will become the active project, as will be shown in the top, blue bar. This keeps the calibration for this capture session. You should now have two project files in the subject's directory.

Create a Template from the First Walking Trial

1. Select the **Motion Capture > Output** panel. Activate the Raw Video (.vc) and Tracked Binary (.trb) check-boxes.
2. Type in a filename (e.g. Walk).
3. Set the duration to be long enough to record one full step cycle.
4. Press the **Record** button. This will produce a **Walk1.trb** file.
5. Next, load the **Walk1.trb** file.
 - Done by either pressing the **Load Last Capture** button in the Motion Capture > Output panel or selecting **File > Load Tracks File**.
 - Loading a file will automatically bring you to the Post Process tab.
6. Select the **Post Process** dashboard, and then press **Quick ID**. An Identifying window appears. Activate the **Rectify** check-box.
7. Identify each marker with the correct name by clicking in the 3D view. The stick figure will automatically be drawn as you identify the markers and will help to highlight mistakes (gaps in data, marker mis-identifying, swaps, ghost markers).
8. Play the trial to make sure it is identified throughout the entire trial. If not, go to the frame you used for identification (usually frame 1) and press **Select Visible Frames**, located in bottom right of screen and press then press **Rectify**. Check again by playing the trial.
9. Gaps in data can be filled by using Join Cubic and/or Join Virtual functions.

10. Press the **Create Template** button.

Figure 2-9. Create Template Interface



The number of data frames should be at least 80% of the total number of frames. Save this **Walk1.trb** file (save over previous file). It now has the correctly identified marker names.

11. Save the tracks file by selecting **File > Save Tracks**.

12. Save project file by selecting **File > Save Project**.

- The template becomes part of the project, yet the project still needs to be saved.

Extend Template

When you create a new template it uses only what is in the currently loaded tracks file to make the full template. When you extend the template, it also just uses what is currently loaded, but it doesn't throw out the existing template information. In effect, you are creating a new template from the original tracks file and the one you currently have loaded that has been combined into a single tracks file.

Extending a template will only increase the allowable range of motion in the linkages and never reduce them.

Collect the Walking Data

- Go to the Motion Capture interface.
 - Collect the Walk trial. You may want to use a naming convention that adds a descriptor of the movement if your subject is doing multiple trials (e.g. Walk1.trb, Run1.trb, etc.).
- If you are collecting force or EMG activity, activate the **Analog (.anb)** check-box.
- Set the capture duration.
 - The duration should be representative of the length of the trial, typically 5-10 seconds. You can set the duration to be the maximum that you would ever expect, and then you could press the **Record/Stop** button for shorter, more typical trials.
- Record the data.

- Press the **Record** button. This will produce a **Walk1.trb** file as well as VCX and ANB files of the same name.
5. View the data.
 - To view analog data while collecting, choose the **2 Panes: Top/Bottom** layout and set the top window to **3D View**. Then set the bottom window to **Analog Display (Data Views > Analog Display)**. You can now view one or all of the analog channels in this window.
 6. Edit the data.
 - Editing is done using the **Data Views > XYZ Graphs**.
 7. Save Tracks (**File > Save Tracks**)

This concludes this quick-start chapter for Movement Analysis applications (Helen Hayes/OrthoTrak marker set). If the Post Processing is to be done in the OrthoTrak software module, this requires TRB files for one static and up to several walking trials. The walking trials may also have their associated ANB files if analog (force and/or EMG) data has been collected.

Quick-Start Tutorial for Animation Production Applications

Topic	Page
Overview	3-1
Starting Cortex	3-1
Project Initialization	3-2
System Calibration	3-3
Marker Placement	3-7
Data Capture	3-8

Overview

This chapter provides a quick reference to begin using your motion capture system for Animation Production applications, and is intended for the more advanced motion capture system user. For Movement Analysis applications, refer to Chapter 2, Quick-Start Tutorial for Movement Analysis Applications.

Note: This Quick-start Guide uses an animation marker set and starts with a project file that is located in the **C:\Program Files\Motion Analysis\Cortex\Samples\Animation Calibration** folder. The basic methodology outlined here can be generalized to other marker sets.

Starting Cortex

1. Turn on the Host computer and login.
2. Turn on the Ethernet switch and CP-8 Power Hub. The cameras will automatically power up.
3. Launch the **Cortex** software by double-clicking the icon located on your computer's desktop.

Project Initialization

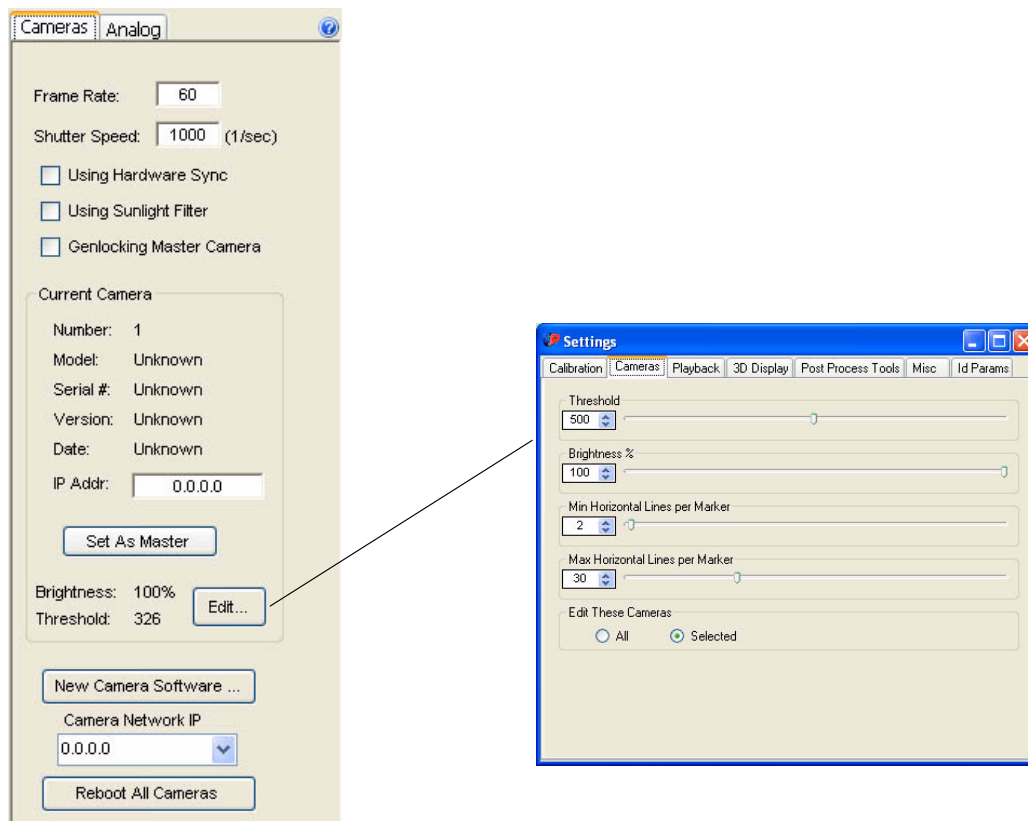
1. Load a previous project (**File > Load Project**) that has an animation marker set.
 - Project files contain information about calibration, thresholds, masks, tracking parameters, marker sets, and templates.
 - By loading a previous project that contains all of this information, you will not have to re-enter it all each time you start a new capture session. You will only need to update the calibration, if necessary.
2. Immediately save the project file in a new folder (**File > Save Project As...**).
 - Create a new folder for the subject and save your project there. This directory now becomes the default **Cortex** directory.

Note:

Make sure that you do not write over previous projects. Separate projects are needed in order to run trials for that particular day. If calibration VC files are written over, then recreating the calibration parameters in Post Process mode will not be possible.

3. In the **System > Cameras** panel, choose **Frame Rate** and set the shutter speed, brightness, and threshold.

Figure 3-1. System > Cameras Panel



4. Press **Connect to Cameras**.

- The first time you do this step, a message indicating that “X number of cameras were found, existing project has 2. Do you want to modify project?” may appear. See [Figure 3-2](#). Press the **OK** button.

Figure 3-2. Connect to Cameras Status Pop-Up (Example)



System Calibration

Note: System calibration should be done at a camera speed of 60 Hz.

Start the Calibration Process Using the (4-Point) Calibration L-Frame

1. Place the calibration L-frame device or four markers (L-shaped) on the floor or on the forceplate.
 - The calibration L-frame markers are set under **Tools > Settings > Calibration > Calibration Frame**. These markers have been placed in a particular orientation and precise distances apart in order to tell the software the origin and coordinate (XYZ) system of the lab/room.
2. Under the **Calibration > Calibrate** panel, activate the **Camera Aiming** check-box.
3. Press the **Run** button.
 - All the number buttons on the bottom of the **Cortex** interface should turn yellow if all of them can see the L-frame.
4. Select **Layouts > 2 Panes: Top/Bottom**.
 - We want the 3-D Display window and 2-D Display window showing. These can be set by left-clicking in the window to make it active and then selecting **Data Views > 3D View** or **2D View**.
5. Check the 2-D views for each camera.
 - Generally, there should only be four markers in each camera view. If there are less, you may need to adjust the view of the camera or you can also adjust the threshold to see more markers. If you have more, you can mask out extraneous data points. To

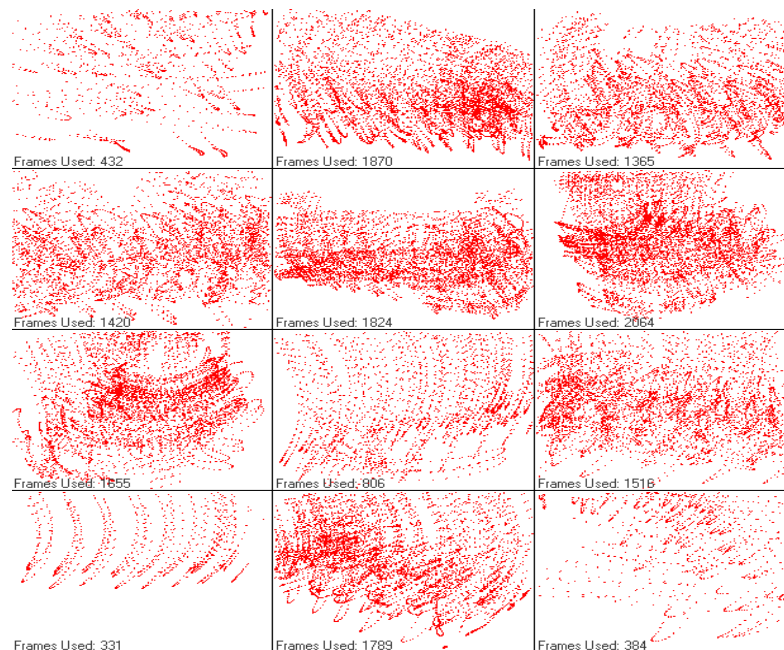
mask, press **Pause**. While in one of the 2-D views, press the middle mouse button and hold it, then drag a square over the bad data to mask.

- To cover all areas in a large capture volume, you may need to aim some of the cameras where they will not see the Calibration L-frame. The **Cortex** software will then calibrate these cameras during the wand calibration procedure.
6. Check the 3-D display and camera locations.
 - If it is not already set, right-click and select **Show Cameras**. All of the cameras should be in the correct place.
 7. Optimize the camera positions and their orientation.
 - New camera positioning should be done at this point if needed.
 - Right-click in the 3-D view and select **Show Camera Field-of-View**. You will probably have to change the length of the field of view to more than the default value of 4000 (4 meters). This adjustment slider is found in the **Tools > Settings > 3D Display** tab. Try 9000 (9 meters).
 - Turn the capture volume on by right-clicking your mouse in the 3-D view and selecting **Show Volume**. This volume is a visual aid helpful in this process of aiming the cameras properly. The volume dimensions are entered under **Tools > Settings > Calibration > Capture Volume** tab in the window.
 - Your camera field of view should cover the desired volume. Try and align edges of the volume box with edges of the camera field of view. This may require that one person moves the camera on 3 axes, while another person directs the movements.
 8. Once the camera position is optimized, press the **Collect and Calibrate** button in the Calibration with Square field. If sound is enabled and you have speakers turned on, you will hear a sound.

Continue the Calibration Process Using the Calibration Wand

1. Remove the L-frame from the capture area. It will need to be completely out of view from all cameras.
2. Set the wand length in the **Calibration with Wand** field.
 - Make sure the wand length is set at 200mm or 500mm depending on the wand used.
3. Set the capture duration.
 - The wand capture duration should be around 60 seconds, or long enough to cover the capture volume. During the 60 seconds, 1/3 of the time should be spent waving the wand parallel to each axis: x, y, and z.
4. Press the **Collect and Calibrate** button in the Calibration with Wand field.
5. Begin waving the wand to cover the capture volume as much as possible.
 - The object of this exercise is to cover the entire capture volume by waving the wand both horizontally and vertically through the cameras field of view. If you look at the 2-D fields of view, you should have only a small amount of white space. The better the coverage, the better the calibration.

Figure 3-3. Proper Wand Calibration Coverage



6. When finished, and the Wand Processing Status window appears, you can select the **Run Again** button. This will recalculate the calibration with continued emphasis on the wand data and will refine all the camera parameters. This step can be repeated several times until the calibration data (residuals) changes very little or becomes worse.

Figure 3-4. Wand Processing Status Window



7. Check the calculated Focal Lengths.
 - The Focal Length for each camera is calculated and should be close to the value that is set on the lenses.
8. Check the 3D Residual Values:

- The 3D Residual values should be small. The Standard Deviation should be approximately half of the 3D residual. Press the **Run Again** button until the values in the calibration processing window stop changing significantly or begin getting larger.
9. When everything looks good and you are ready, press **Accept**. If the calibration still does not meet the desired values, you can press the **Reject** button. You may have to do one part or all of calibration again.

Note: It is possible to calibrate with previously collected files.

10. Save the project (**File > Save Project**).
- When you press **Accept** in the step above, you will get two messages stating “Calibration has been saved”. This message indicates that the project is saved to a system folder. You need to select **File > Save Project** in this step, since the system folder will be overwritten each time a calibration is done.

Possible Problems with Calibrations—How to Solve

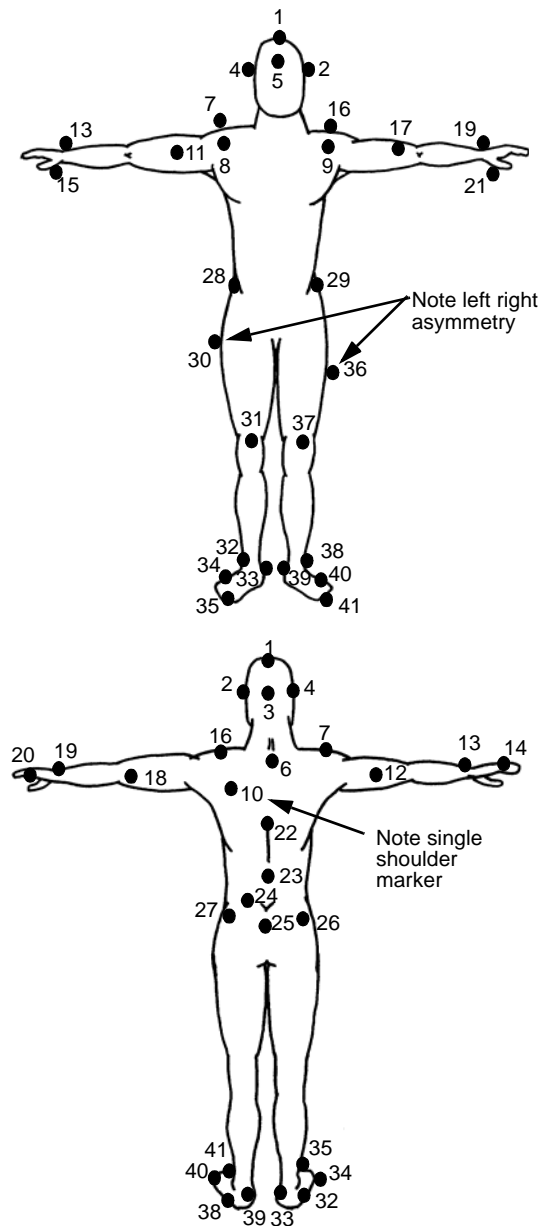
- Wrong placement or measurements of the calibration L-frame. Verify all measurements and x, y, and z axes that are set.
- Check the brightness of the cameras and the use/non-use of masks. Remember to limit the use of masks and make them as small as possible if they are in line from the camera through the intended capture volume. If any markers go through a masked area, the data will be ignored.
- Too many extra marker images are possible causes for a bad calibration. Watch out for anything reflective such as extra markers, reflective material on shoes, shiny floors, debris in carpeting, and sunlight coming in through windows.
- If calibration problems persist, contact support@motionanalysis.com

Marker Placement

Note: For the purpose of this illustration, this example uses a typical animation marker set. The theory may be extended to different marker sets.

1. Attach reflective markers following the markers listed in [Figure 3-5](#). Placement on bony points is ideal if available. Consult an anatomy book as reference for palpating these points.

Figure 3-5. Typical Animation Marker Set



Note-When placing markers on end segments, the markers should not form a line and should not have mirror symmetry. Thus, thumb and hand markers should never be the same distance from the wrist marker and should be well separated.

Head and Neck

1. TopHead
2. L_Head
3. B_Head
4. R_Head
5. F_Head

Arms and Hands

11. RBicep
12. RElbow
13. RWrist
14. RPinky
15. RThumb
17. LBicep
18. LElbow
19. LWrist
20. LPinky
21. LThumb

Back and Root

22. MidBack
23. LowBack
24. RootOffset
25. Root

Shoulders and Sternum

6. TopSpine
7. RShoulder
8. FRshoulder
9. FLshoulder
10. ShoulderOffset
16. LShoulder

Pelvis and Hips

26. BRHip
27. BLHip
28. FRHip
29. FLHip

Legs and Feet

30. RThigh
31. RKnee
32. RAnkle
33. RHeel
34. RMidfoot
35. RToe
36. LThigh
37. LKnee
38. LAnkle
39. LHeel
40. LMidfoot
41. LToe

Data Capture

Load an Animation Marker Set

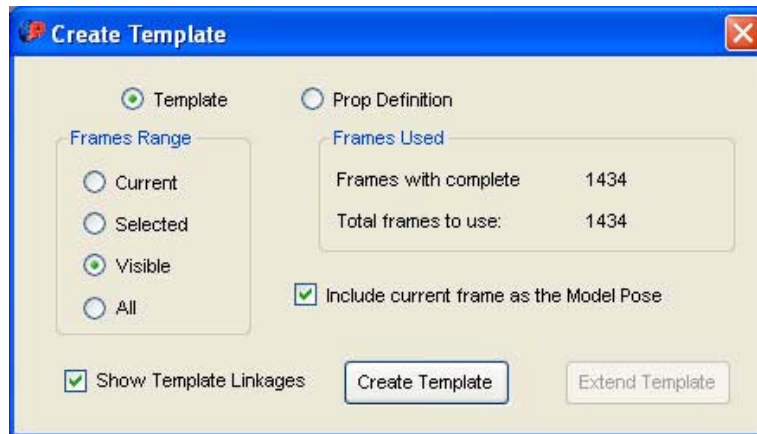
1. Select **File > Load Marker Set**.
 - Load the **MarkerSetBody.prj** file that contains the animation marker set.
2. Save the project file (**File > Save As Project**) with a new name (i.e. **Dave.prj**), which will become the active project, as will be shown in the top blue bar. This keeps the calibration for this capture session. You should now have three files in the project folder.

Create a Template from the First Walking Trial

1. Select the **Motion Capture > Output** panel and activate the **Raw Video (.vc)** (if you're doing a live capture) and the **Tracked Binary (.trb)** check-boxes.
2. Create a range-of-motion (ROM) file. Type in a filename (i.e. DaveROM) and set the duration to be long enough to record one full step cycle. If you're using the example data, the program will know how long to record. For more information on the ROM files, refer to [“Building a Template from the Range of Motion Trial” on page 9-5](#).
3. Press the **Record** button. This will produce a **DaveROM1.trb** file.
4. Load the **DaveROM1.trb** file.
 - Done by either pressing the **Load Last Capture** button in the **Motion Capture > Output** panel or selecting **File > Load Tracks File**.
 - Loading a file will automatically bring you to the Post Process tab.
5. Select the **Post Process dashboard** and then press **Quick ID**. An Identifying window appears. Activate the **Rectify** check-box.
6. Identify each marker with the correct name by clicking in the 3D view. The stick figure will automatically be drawn as you identify the markers and will help to highlight mistakes (gaps in data, marker mis-identifying, swaps, ghost markers).
7. Play the trial to make sure it is identified throughout the entire trial. If not, go to the frame you used for identification (usually frame 1) and press **Select Visible Frames**, located in bottom right of screen and press then press **Rectify**. Check again by playing the trial.
8. Gaps in data can be filled by using Join Cubic and/or Join Virtual functions.

9. Press the **Create Template** button.

Figure 3-6. Create Template Interface



10. Save the tracks file by selecting **File > Save Tracks**.
11. Save project file by selecting **File > Save Project**.
 - The template becomes part of the project, yet the project still needs to be saved.

Start Collecting Motion Data

You are now ready to collect data for this subject.

Planning a Motion Capture Session

Topic	Page
Overview	4-1
Studio or Lab Preparation	4-1
Prior to the Capture Session	4-2
Job Assignments and Tasks During the Session	4-3
Additional Equipment	4-4
Motion Capture Terminology	4-5
Motion Capture Session Sequence of Events	4-6
Capturing the Data	4-7

Overview

The motion capture process starts by collecting raw video data of the subject. The success of the final motion data will depend not only on the quality of the subject's performance but also on the organization skills and experience of the **Cortex** operator. The quality of the **Cortex** data can be greatly affected by the events leading up to and during the motion capture session.

An efficient motion capture session can ultimately save time and money. Although this chapter is geared towards animation, some information may be helpful for both animation and biomechanics. What follows are suggestions that can help make the motion capture session run smoothly.

Studio or Lab Preparation

At least a day before the capture session, the **Cortex** user should know the capture volume required and the nature of the motion capture project. This information is essential for an efficient motion capture session. Knowing the capture volume allows for the advanced selection of the appropriate marker size for the session.

It may be appropriate to use different capture volumes for the different moves of a capture session. Changing the capture volume size and optimizing this volume could take up to one hour, so this switchover should be scheduled during a break. The approximate volumes can be set up ahead of time using tape on the floor to mark the capture volume boundaries and the position for the tripod legs (if used) of each camera.

If more than one subject will be performing in a capture session, it is a good idea to mark out a capture volume practice area away from the actual capture area. This will allow the next subject to practice before motion capture.

On the **Cortex** host workstation, create the appropriate directories and project files. For batch edit work in Post Processing, a separate file folder for each project and its associated capture files is strongly suggested.

Make sure there is enough room on the **Cortex** workstation's hard disk. If you know the number of trials you are going to capture and the approximate length of each trial, you can estimate the amount of hard disk space you will need. Use some form of backup medium (e.g. CD-ROM, Zip disk) to back up previous data and clear space on the hard disk for the new trials.

Prior to the Capture Session

Several days prior to the capture session, schedule a visit by the subject and any producers or directors involved in the motion capture session. If the subject has not worked with reflective markers, this will allow time to become familiar with marker placement and to practice in the marked-out capture area.

You will want to specify the most desirable type of clothing for the session. Remember, your goal is to capture the fine details of the movement of the body, not the movements of clothing on the body. The rule is to apply markers to skin whenever possible. The areas on the body that present the greatest potential problems are shoulders, the rear neck, sternum, mid back, and the root. A tank-top shirt may be used to expose the shoulder, neck, and sternum. The root marker should be placed low on the spine in an area below the belt line where there is usually very little clothing movement. If a mid back marker is used, the shirt should be rolled up and taped to expose the back. If skin cannot be exposed for all marker placements, then tight fitting clothes or a motion capture body suit should be worn.

Have the subject perform some of the motion capture moves within the capture area. If markers have been placed on the subject, go ahead and capture some data. This would be an ideal opportunity to use a stopwatch to time the duration of each move. These trials can give the **Cortex** user an indication of potential tracking problems, and if this data is taken all the way to the animation software, it will allow the artist to see how well the data fits their models.

Finally, instruct the subject to speak up during the motion capture if there are any problems with the markers. If markers become loose, they will need to be reconnected more securely with tape or rubber bands.

Job Assignments and Tasks During the Session

Director

The director ensures that everyone involved is prepared for the capture session and controls the session, including instructing and critiquing the subject's performance.

Camcorder Operator

The video recording from the camcorder can be very important documentation to aid in choosing the best takes. A video recording may also be useful for post-production promotions. An optional reference video capture is available. Refer to [“Digital Video Option \(EVaDV Software\)” on page 6-21](#).

The camcorder should not be allowed to run during the entire capture session. This means that someone should be assigned to start and pause the camcorder for each take.

The camcorder operator should slate each of the takes. This involves recording the take number, the **Cortex** trial name and number, and any other relevant information on a slate board. After starting the camcorder and before each take, the slate board is held up in front of the camcorder for a few seconds. The audio on the camcorder can also be used. When the slate is held up in front of the camcorder the operator can say the take number, the **Cortex** filename, and any other information necessary. When the take is complete, the director can make audio comments on the quality of the take.

Scribe

Someone should be assigned to take notes and to fill in the Motion Capture Log. A sample of this log is found in [Appendix M, Useful Blank Forms](#).

The take number, **Cortex** file name, and the duration of the take and any comments from the director or subject should be recorded. The scribe can also do a time check using a stop watch to get the length of data capture for each new move.

Cortex Operator

The **Cortex** operator must make certain that the motion capture data is clean and trackable. The **Cortex** operator must make sure the camera calibration is good and that raw calibration data is collected at various times throughout the session as insurance. This is particularly important when there are several people around and a camera might get bumped accidentally. The operator should watch for reflections, changing light conditions, such as sunlight coming through a window, or other external variables which may affect a capture.

Additional Equipment

Props

If the motion capture session requires the use of props, this must be known to all parties well in advance. The type of prop and its use are very important because reflective markers may have to be attached to the prop as well as the subject. One marker may be used to track position, but as many as three markers may be required to show all rotations of the prop. Many props that would ordinarily seem simple become very difficult to deal with during a motion capture session. A good example is the use of a ball as a prop. If a small ball is only being held, one marker may be used to track position. If a large ball is being bounced, three markers may be required to show all rotations.

Reflective or glossy material should **not** be used in the construction of props, and very large props may occlude the subject's markers. Remember, the design of the prop and how it affects the subject's movement are more important than the prop's physical appearance. Props may also be assigned separate templates (see [“Multiple Tracking Objects” on page 9-7](#)).

Camcorder

Used to completely document each trial, a camcorder will allow the producer to rank the trials of a move and will also give the animation artist something to use as a reference for the completed animation. See also [“Digital Video Option \(EVaDV Software\)” on page 6-21](#).

Still Camera

Photographs of the subject, with markers attached, will help the artists understand the correspondence between the marker data and the actual figure.

Slate Board

A slate board and chalk or grease-pen board will provide an easy way to relate the camcorder record to the **Cortex** data.

Markers, Tape, Pre-tape Liquids, Rubberbands

An adequate supply of reflective markers, double stick tape, paper tape, and “Tuff Skin” or “New Skin” should be available. For rough and tumble sessions, the best method to adhere markers is by using Velcro™ on a skin-tight motion capture suit. However, markers can be applied directly the skin.

If markers must be placed directly on the skin and the subject will be performing athletic moves in which perspiration might be a problem, pre-tape liquids like “Tuff Skin” or “New Skin” can help make double-stick tape adhere better. These products must be applied to dry skin and allowed to set for a minute or two before the marker is attached.

Rubberbands looped around the marker and limb also work well to stabilize the markers. Rubberbands can be looped together to increase diameter and prevent restriction of blood flow. Rubberbands can be used around the elbows, wrists, hands, knees, ankles and toes.

Backup Media

Spare CD-ROMs, Zip disks, or some other backup medium should be available for backups and data transfers.

Motion Capture Body Suit	A motion capture body suit with Velcro attachments for markers provides a quick way to prepare a subject for motion capture. The use of the body suit is especially effective when subjects are involved in rough or contact-type motion capture sessions, common in animation applications.
Camcorder Tapes	Depending on the length of the capture session, spare video tapes should be on hand.
Music Player	Either a CD or tape player can provide musical accompaniment. Music helps calm and smooth out the subject's performance not only with dance, but athletic moves as well.
Stop Watch	A stop watch is handy for calculating the duration of each new move.
Sample Form	<p>You should decide at the outset whether you will build a hierarchical skeleton. If you decided to, there are two software methods available: SkB (Skeleton Builder) and Calcium/Si.</p> <p>Blank forms to help you define skeleton parameters for each of these methods can be found in Appendix M, Useful Blank Forms. You may want to copy one of these forms for recording your project measurements.</p>

Motion Capture Terminology

Some terms that are useful to a motion capture session are **moves**, **trials**, and **takes**.

Move	A move is an event or routine performed by the motion capture subject. A move can be as simple as a neutral stance position, or as complex as a 2 person, 30 second dance routine. The director and subject will work from a move list .
Trial	Multiple trials of a move should be taken. The number of trials depends on the complexity of the move, the subject's performance, and quality of the Cortex raw data. Usually, three trials per move is adequate. It is important that the director or subject's comments about the quality of the trial (which trial was the best) be recorded on the Motion Capture Log. Knowing which trial of a move is the best will allow the Cortex user to track only the best trial.
Take	A take is the master number used to relate what is on the camcorder's video tape to the Cortex filenames and trial numbers. The take number is displayed on the slate board and on the Motion Capture Log. Every new image recorded on the video tape should have a new take number. This should include calibration collection, initialization and T-pose/Init pose stance positions. You should never re-use or redo a take number. If a data collection is aborted for some reason, e.g. a marker fell off, then the Cortex filename and trial number can be overwritten, but the take number should change.

Motion Capture Session Sequence of Events

The Day Before:

1. Optimize the camera positions and orientation to the capture volume.
2. Calibrate the volume of the capture area.
3. Determine the correct marker size to use. **Cortex** raw data should show 2 lines or greater per marker.
4. Setup the **Cortex** project with the correct markers, virtual markers, linkages, segments, etc.
5. If possible, collect and track the markers on a person to verify that the tracking parameters are optimal.
6. Verify that there is enough space on the workstation's hard disk. If there is not enough space, back up the previous files and then erase them from the hard disk.
7. Organize the markers, tape, and props to most efficiently facilitate the session.

The Day of the Motion Capture

Before the subject arrives:

1. Load the **Cortex** project.
2. Optimize the threshold settings.
3. Collect calibration data sets (both seed and wand).

Subject Preparation

1. Ensure subject's clothing is appropriate.
2. Allow the subject to warm up.
3. Attach the markers according to predetermined placements.

Note: Asymmetrical marker placement on the subject is critical for obtaining the best marker data.

4. With the markers in place, take still photos of the subject from the front, side, and rear view.

Note: If you are taking photographs, do not use the flash attachment on the still camera while you are collecting data. A flash during data collection can corrupt the data.

5. Allow the subject to practice in the capture volume with the markers on.
6. Prepare for the calibration collection. Explain to everyone the importance of not bumping the camera tripods.

Capturing the Data

Calibration

Note: This section provides a general overview of the calibration process. For complete calibration information, refer to Chapter 8, Calibration Panel.

Collect the Square (Seed) Calibration

1. Fill out the Motion Capture Log and slate board for the first square (seed) calibration. This would be Take 1 and an **Cortex** filename, for example “**CalSeed**”.
2. Prepare the **Cortex** system for data collection. Press the **Collect and Calibrate** button to trigger the event button.
3. Verify that the camera buttons turn yellow after the Seed calibration is complete.
4. Remove the calibration seed device (calibration L-frame) from the capture volume.

Collect the Wand Calibration

For best results it is recommended that you collect and use wand calibration data.

Prepare the **Cortex** system for wand calibration. The duration of the wand calibration is directly correlated to the capture frame rate. A typical duration for a small capture volume is 30 to 60 seconds. Large volumes with ten or more cameras can take 120 to 180 seconds, and very large volumes may take up to 240 seconds.

Collect and verify that the wand calibration data is good. It may be necessary to reposition or move cameras and to retake both the seed and the wand calibration data if one or more cameras has large areas without wand calibration data. You can continue pressing **Run** after it finishes, until the calibration numbers stop changing.

Your wand data should cover the entire capture volume. A common method of ensuring better wand data is to use a 1/3 method. That is, hold the wand markers in alignment along each axis (X, Y, and Z) for 1/3 of the wand capture session.

Collecting Trial Data

Subject Initialization

The type of subject initialization depends on the application.

- In animation applications it is the “T-Pose” or “Init Pose” trial
- For **OrthoTrak** it is the Static Trial
- For **KinTrak** it is the Neutral Trial

In general, the procedure is as follows:

1. Have the subject stand in the capture volume with the markers on. On the **Cortex** system. Look for any reflection and light source that might interfere with the capture and correct the problem.

2. Prepare the **Cortex** system by entering a filename. If the subject's name is Jane use something like JaneInit. For this initialization file, use a duration of 2-3 seconds. The take number should be set to 3. Click on **Collect** to arm the event button. You can enter a long duration (e.g. 20 seconds) and then press the hand-held event button a second time at the end of the move.
3. Update the slate board with the new take number and **Cortex** filename. Make the same entries in the Motion Capture Log.
4. When everybody is ready, the director can say "roll video", then slate the video. Now the **Cortex** operator gives the signal for the subject to start and presses the event button to start the data collection. The event button must be pressed a second time to stop data collection at the end of the move.
5. For initialization, use the T-pose/Init pose. In this pose the subject faces forward and raises both arms straight out from their sides with the thumbs oriented up. Perform this motion within the duration of the capture time.
6. Pause the camcorder.
7. Collect two trials of this "initialization" move.

Capturing the Moves

1. Before capturing, have the subject practice each move.
2. Enter a duration **longer** than the estimated length of the move.
3. Enter an **Cortex** filename, the duration of data capture, the trial number and trigger the Event button.
4. Update the slate and the Motion Capture Log.
5. The director should ask if everybody is ready and then say: "Roll video", "Slate video."
6. The **Cortex** operator presses the event trigger button and the subject begins the trial. When the trial is finished, the event trigger button is pressed again to complete the capture.
7. Comments on the quality of the trial should be entered into the Motion Capture Log and on the audio of the camcorder.
8. The **Cortex** operator quickly reviews the raw data and looks for any problems.
9. This process is usually repeated for 3 trials of each move that is scheduled.

Note: Usually only one trial is tracked. The other trials are there for insurance and to allow the end user to pick the best trial.

Collecting Calibration "Insurance" Data

For insurance, it is a good idea to periodically collect raw calibration data whenever there is down time. As the number of people increases in the capture studio, the chance for bumping a camera (if tripods are being used) increases and "insurance" calibration data suddenly becomes very valuable.

Wrapping It Up

After all the trials have been collected, perform the following to wrap up the process:

1. Collect the last calibration trial.
2. Backup all the **Cortex** trials on a CD-ROM, Zip disk, or other backup medium. Label and store the tape in a safe place.

3. Remove the video tape from the camcorder and set the safety tabs on the tape to prevent being recorded over.
4. Consolidate and make copies of the motion capture logs and forms.
5. If necessary, give the video tape and Motion Capture Logs to the director so the best trials of each move can be indicated.
6. Place logs and offset forms in a binder. Clear plastic inserts can be added to hold the still photos. The binder along with the video tape will provide important information to both the **Cortex** user tracking and editing the data, and for the artists who will apply the final data to the model.

Camera Setup

Topic	Page
Setting Up a Motion Capture Laboratory	5-1
Deciding On the Optimum Number of Cameras	5-2
Setting Up the Cameras	5-11
Eagle Camera Physical Dimensions	5-17
Hawk Camera Physical Dimensions	5-18
Hawk-i Camera Physical Dimensions	5-19
Overview of the System Calibrating Process	5-20
Placing the Calibration L-Frame	5-22
Marker Sizes and Maximum Distances for Eagle/Hawk Cameras	5-23
Troubleshooting Eagle and Hawk Camera Problems	5-24
Relationship Between Capture Volume and Marker Size	5-29

Setting Up a Motion Capture Laboratory

Camera placement is the most important aspect of setting up your motion capture laboratory. If properly done, good camera placement will reward you with highly accurate and consistent results, and greatly reduced editing time.

Optimum Laboratory Conditions

- Fluorescent lights are the best ambient light when red or notch filters are used on the motion capture cameras.
- Windows should be covered with curtains to eliminate direct outside light.
- Carpeting or other non-shiny floor surfaces are preferable to tile flooring which can reflect opposing ring lights.

Recommended Supplies

- A stepladder—for adjusting the cameras/tripods.
- Masking tape—to mark the floor when measuring the capture space and setting up the cameras.
- Reflective markers—to attach to the subject and also enough to place on the floor when adjusting the cameras.
- Other supplies include surgical tape, electrode collars for applying markers to people and gaffer's tape (black masking tape).

Deciding On the Optimum Number of Cameras

There are several objectives to consider when deciding how many cameras should be used and where they should be placed.

1. There should be a sufficient number of cameras to insure that, at all times, all markers will be visible by at least two, and preferably three, cameras. In general, the number of cameras must be increased when:
 - the motion of the subject becomes less constrained
 - the number of subjects or objects increases
 - the capture volume increases
2. As more cameras are used, each camera should view only a portion of the capture volume to achieve higher accuracy and prevent too many cameras from seeing any one marker. For large capture volumes with a large number of cameras (10+), it is recommended that all 4 markers on the calibration L-frame are seen in only 1/4 to 1/2 of the cameras. You can then use the Extend Seed function (see [“Extending the Seed Calibration” on page 8-18](#)) to calibrate the remaining cameras.

Note: When more than 5 or 6 cameras see the same marker, the accuracy of tracking is not increased and computation time increases.

3. Camera views should not include areas outside the capture volume to ensure the highest possible spatial resolution.

The number of cameras in a typical motion capture setup can be as few as 2 or as many as 250. The following provides some guidelines for deciding on the number of cameras to use. In the following figures, all measurements are in meters.

6 Cameras

For motion capture involving only one subject, where the occlusion of markers is not a problem, six cameras may be adequate. This configuration is often used for gait analysis and other similar biomechanical applications. The two end cameras are often tilted so that the long axis of the view areas is vertical. For optimum results, all cameras should be about 2.5 meters above the floor. See [Figure 5-3](#).

8 Cameras

As a wider range of motion is allowed, the probability of markers being occluded increases to the point that eight or more cameras are required. This is the minimum recommended configuration for animation applications. Cameras should be about 3 meters above the floor. See [Figure 5-4](#).

10 Cameras

In an elongated capture space, ten cameras may prove beneficial. The first 8 cameras should be placed about 3 meters above the floor as in the 8 camera setup. The two additional cameras (9 and 10) should be placed 5 meters above the floor at each end of the long dimension of the capture volume and will probably have longer focal length lenses than the other cameras. See [Figure 5-5](#).

12 Cameras

As the capture volume becomes more elongated, twelve cameras may be required. The first 8 cameras should be placed 3 meters above the floor. Cameras 10 and 11 should be placed as 5 meters above the floor on the ends of the capture volume but closer to the center than cameras 9 and 12

(same height). Therefore, one end of the long volume will be covered best by cameras 9 and 11, while the other end will be covered best by cameras 10 and 12. See [Figure 5-6](#).

14 Cameras

When the sides of the capture volume are too long to be adequately covered by four cameras on each side, an additional pair of cameras with wide angle lenses can be placed in the center of each long side of the capture volume. The first 8 cameras should be 3 meters above the floor. Cameras 9 through 14 should be 5 meters above the floor. See [Figure 5-7](#).

16 Cameras

To use more than 14 cameras effectively, it is usually necessary to break the capture volume into two overlapping sections across the long axis. Every camera must see at least one of the squares in its entirety. All cameras should be placed at least 3 meters above the floor. See [Figure 5-8 on page 5-9](#).

For more information and an example, refer to “[Overview of the System Calibrating Process](#)” on page 5-20.

More than 16 Cameras

As capture volumes increase in size, more than 16 cameras may be required. It is best to consider the capture volume as two or more overlapping regions. For large square shaped capture volumes, up to 32 cameras can be used with the space broken into four regions. See [Figure 5-9](#).

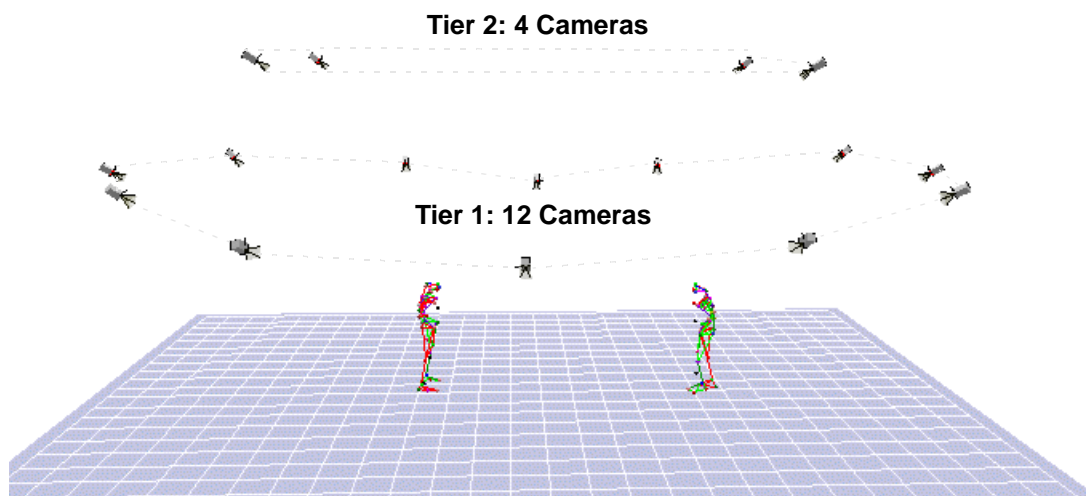
Capture Volumes Between Eagle and Hawk Cameras

Eagle and Hawk cameras use the same high-powered ringlights and have the same limits for marker distances. The difference between the two cameras is that you can use smaller markers with the Eagle cameras (about one-half the size of the Hawk markers). A typical example would be 1/4 inch (6 mm) markers for Eagle cameras and 1/2 inch (12.5 mm) markers for Hawk cameras.

16 Camera, Two-Tier Setup

For an example of this 16 camera, two-tier setup, open the **MAS_16Camera_2Tier.prj** file in the following directory:
C:\ProgramFiles\MotionAnalysis\Cortex50\Samples\LargeVolumes
Also refer to [Figure 5-1](#).

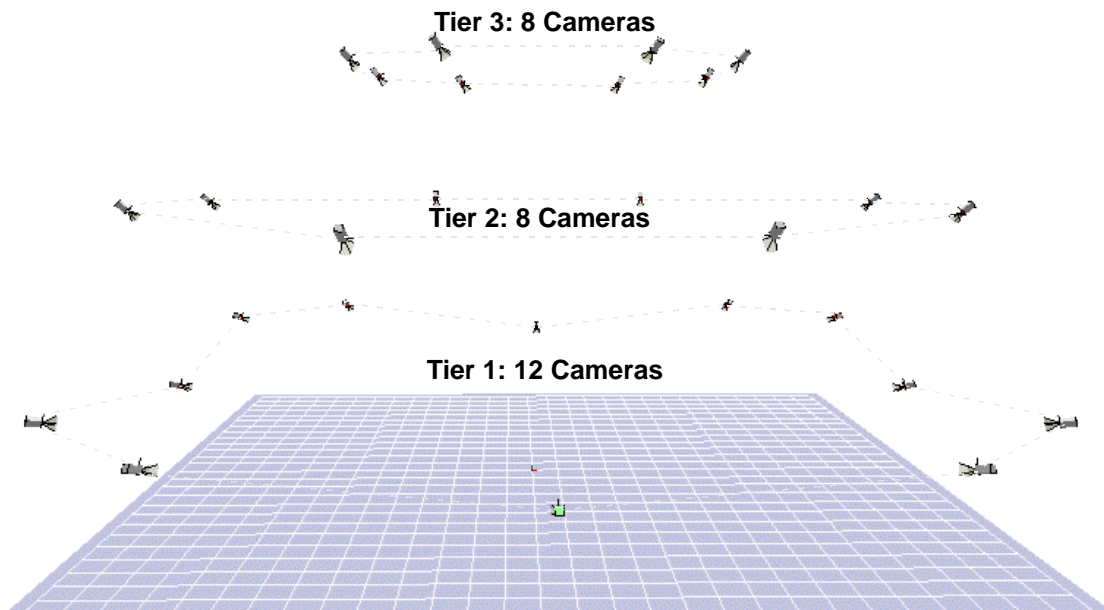
Figure 5-1. 16 Camera, Two-Tier Setup



28 Camera, 3-Tier Setup

For an example of this 28 camera, three-tier setup, open the **Spectrum_28Camera_3Tier.prj** file in the **C:\ProgramFiles\MotionAnalysis\Cortex50\Samples\LargeVolumes** directory. Also refer to [Figure 5-2](#).

Figure 5-2. 28 Camera, 3-Tier Setup



Ideal Capture Volume Sizes

Calculating the ideal volume size for a specific camera setup can have many factors involved and it can become a very hard question to answer. However, a good starting point would be to assume a two person, full body capture area in an ideal space (no restrictions on camera placement, etc.). For this we suggest the figures listed in [Table 5-1](#).

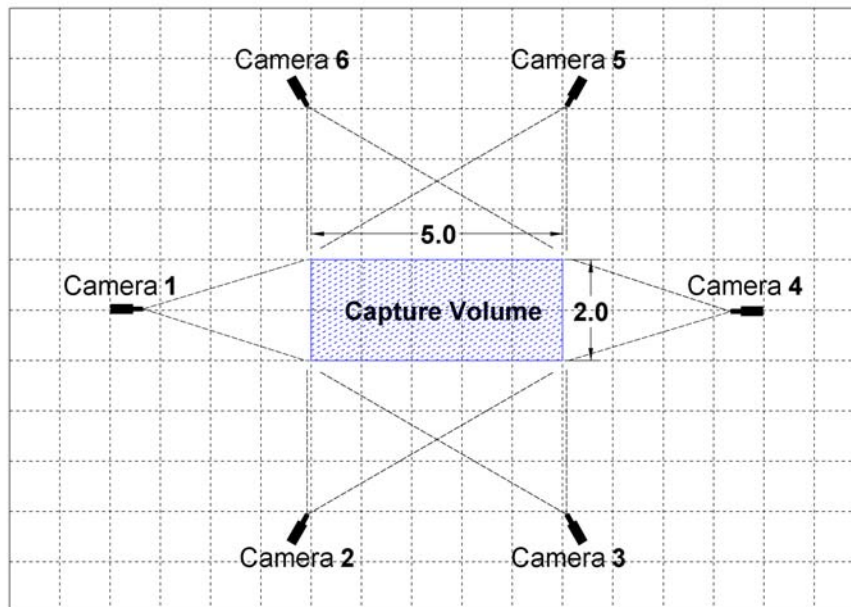
Table 5-1. Ideal Volume Sizes for Specific Eagle and Hawk Camera Setups (with standard lenses^a)

Number of Cameras	Dimensions (m)	Area (m ²)
6	5 x 2	10
10	7 x 5	35
14	9 x 6	54
16	13 x 6	78
32	13 x 11	143

a. Standard lenses are 18-55 mm Zoom for Eagle cameras, and 6-15 mm Zoom for Hawk cameras.

So how is it that 16 cameras give you 4 times the capture area of 8 cameras (you might ask)? Mostly because when using a small number of cameras you end up wasting a lot of the usable viewing cone of each camera. Using more cameras allows for more efficient usage of each individual camera.

Figure 5-3. Typical 6 Camera Setup



Note: Capture volumes may vary depending on room size and the distance from the camera to the capture area.

Figure 5-4. Typical 8 Camera Setup

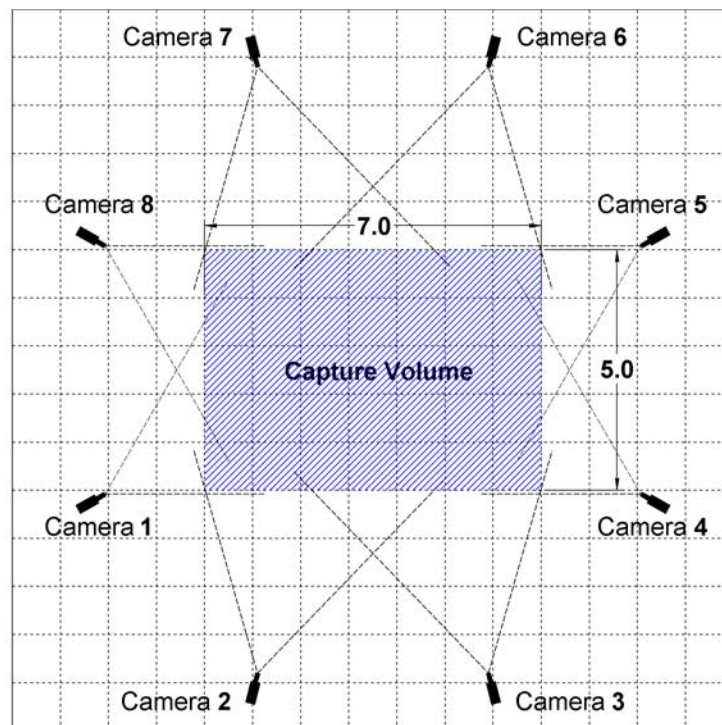


Figure 5-5. Typical 10 Camera Setup

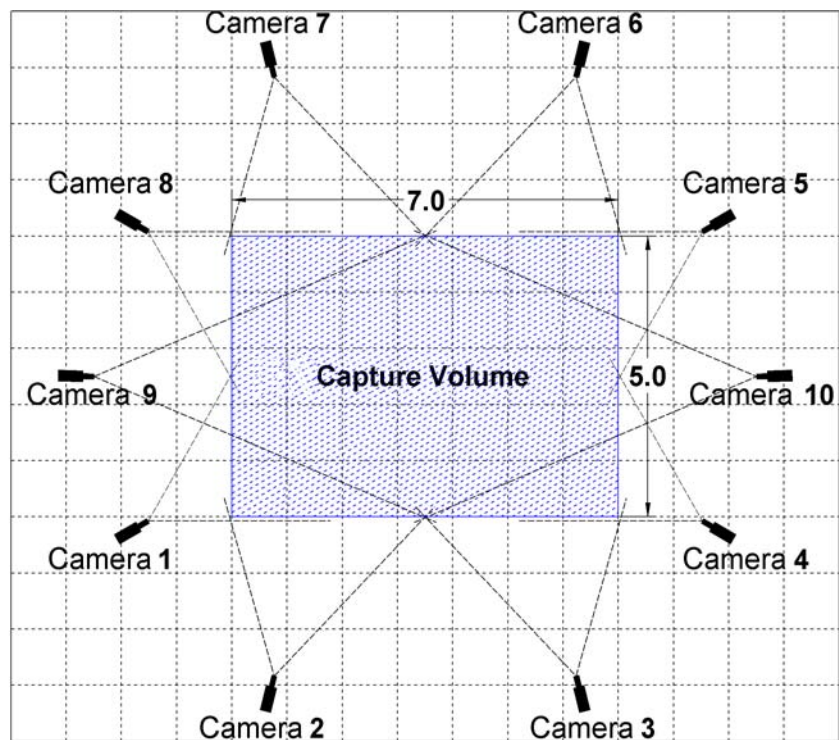


Figure 5-6. Typical 12 Camera Setup

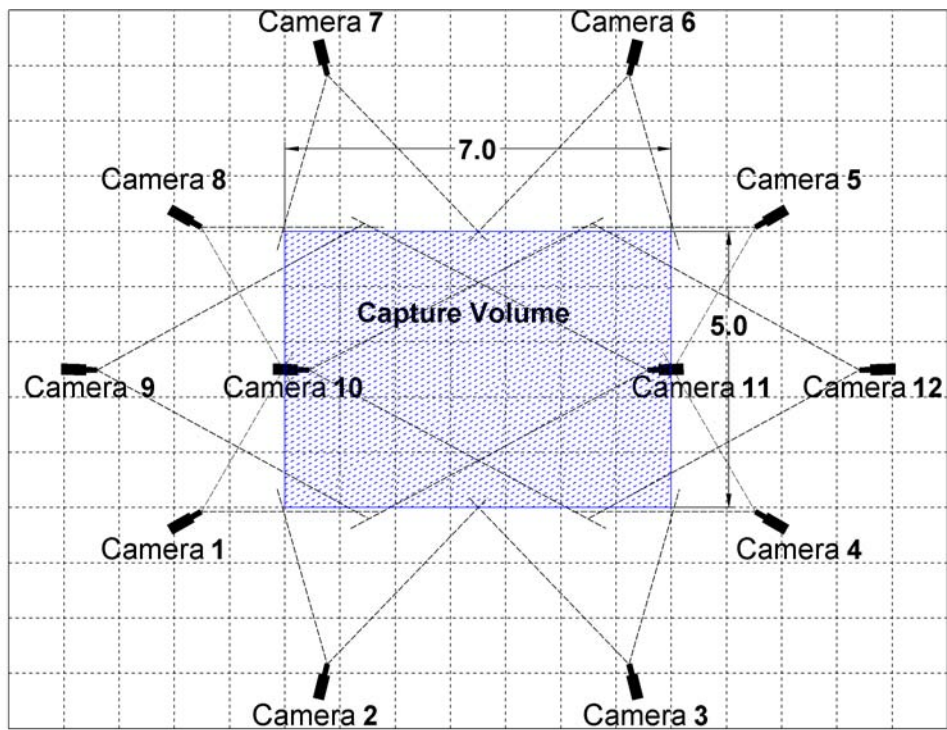


Figure 5-7. Typical 14 Camera Setup

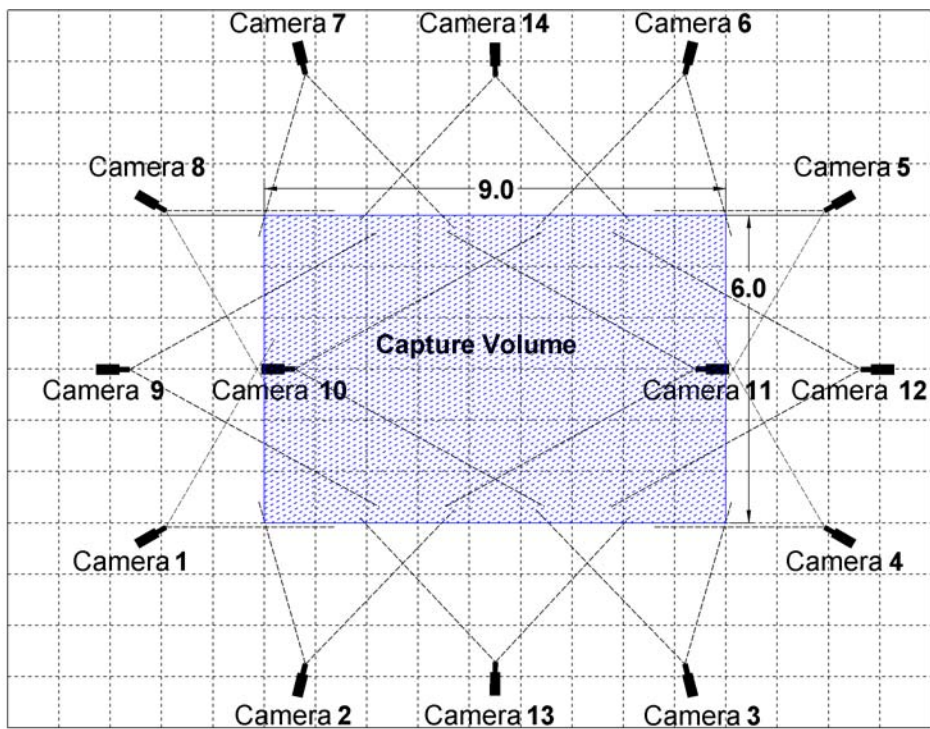


Figure 5-8. Typical 16 Camera Setup

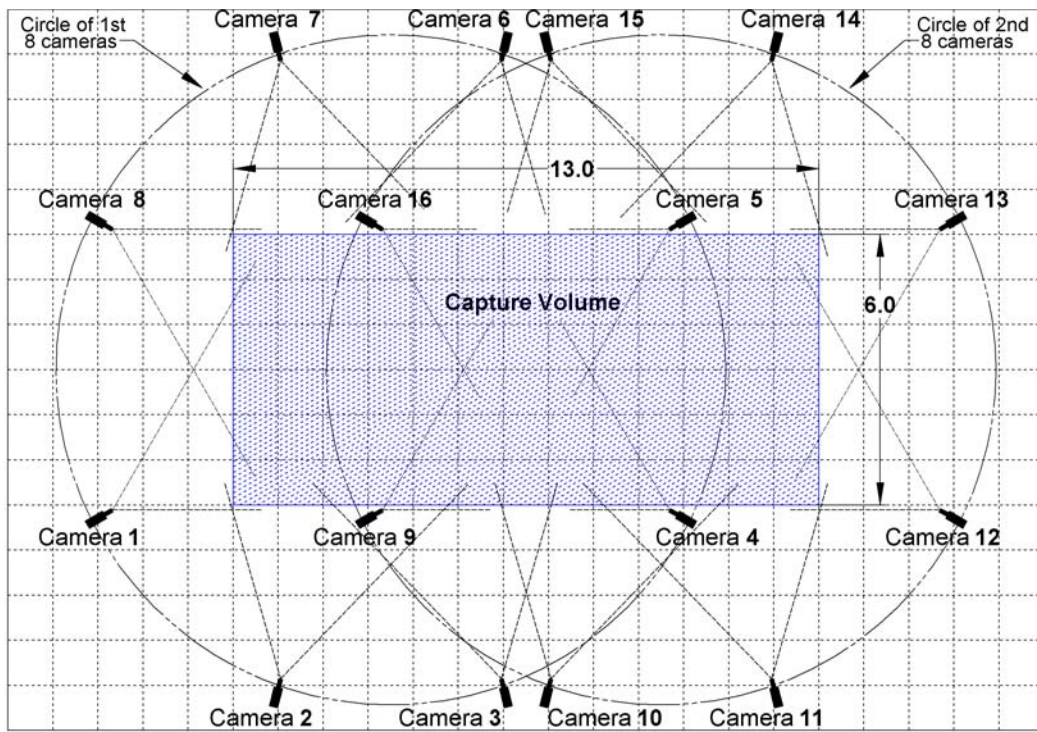
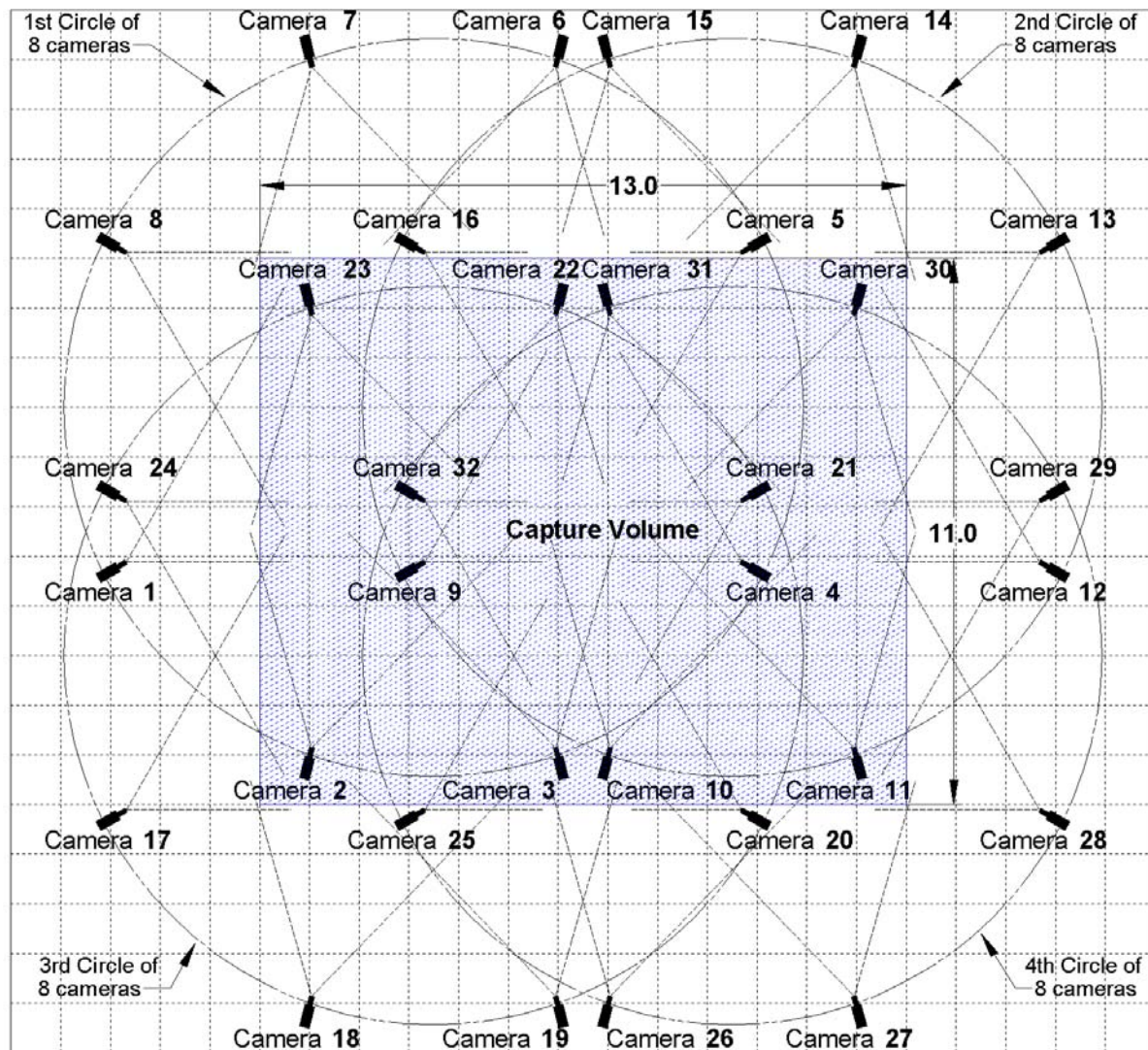


Figure 5-9. Typical 32 Camera Setup



Note - The capture volume is divided into 4 overlapping regions, each containing 8 cameras. The regions are as follows:

Upper left quadrant	Cameras 1-8
Upper right quadrant	Cameras 9-16
Lower left quadrant	Cameras 17-24
Lower right quadrant	Cameras 25-32

Setting Up the Cameras

The key to placing cameras around the capture area is to position them where they will yield the highest resolution without excluding any part of the adjacent capture volume. In other words, if you plan to track 2 gait cycles, do not set up an area suitable for 4 gait cycles. When the tracking volume is increased, the quality and accuracy of the tracking data will decrease.

First, you will want to measure the room to establish the center of the tracking area.

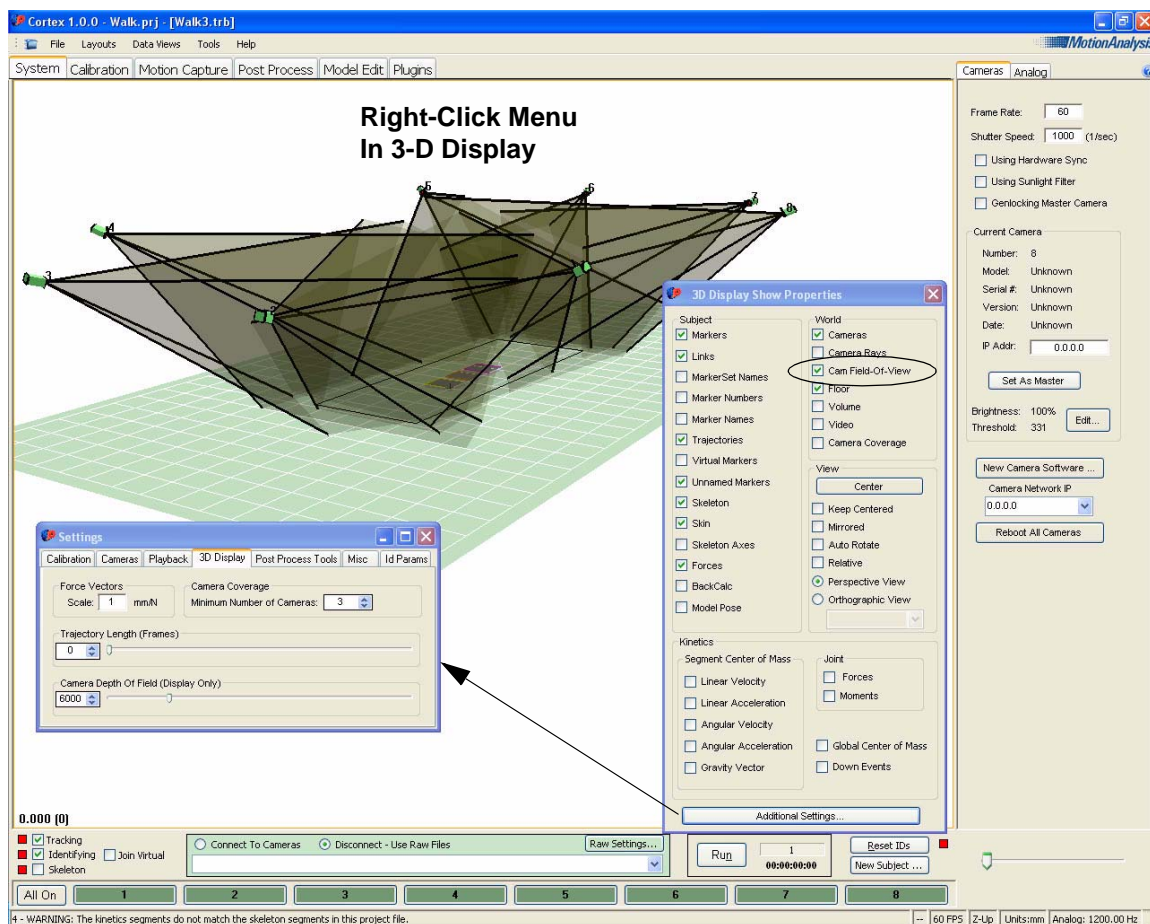
An Example Eagle or Hawk Camera Setup

For example, if you have a 10 x 15 meter room and you are using 8 Eagle or Hawk digital cameras:

1. Measure in from the walls 5 and 7.5 meters. This should be the center of a 10 x 15 meter room.
2. Mark the center of the room or tracking area with a piece of masking tape.
3. Find the corners of the actual capture volume. For optimum tracking, the length and width of the capture volume should be no more than about half the room dimensions.
4. Position the cameras evenly around the capture area. Place the cameras above the top of the capture space, looking down, to prevent cameras from seeing an opposing camera's ring light.
 - For most gait analysis installations, a height of 2 meters should be sufficient. For a larger capture area (e.g. full body or sports analysis), the cameras may need to be raised higher.
 - If a camera must view an opposing camera, use the mask function in **Cortex** to block the offending image. Refer to [“Creating and Clearing Masks” on page 7-9](#).
 - Position the cameras so that they are equally spaced when viewed from the center of the capture area.
 - Beware of making the capture area too large. The resolution and the quality of the data may be compromised.
5. Place cameras in lower positions. Good results can come from adding cameras or positioning the cameras low (on the floor), looking up at the capture subject. This is also effective for capturing markers as the subject is stooped over or lying on the floor. Opposite camera ring-lights can be masked out if necessary.

6. One by one, adjust each camera so it sees as much capture area as possible.
7. To see the camera view, right-click in the 3D view and then select **Show > Show Camera Field of View**. See [Figure 5-10](#).

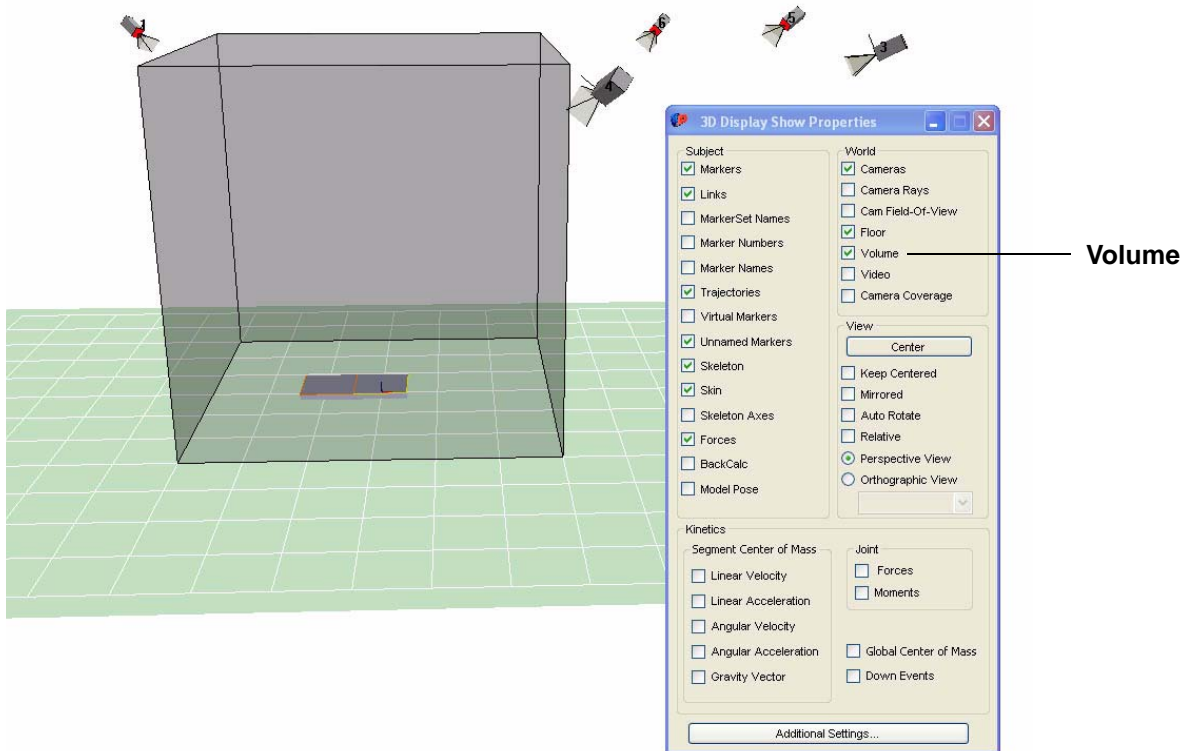
Figure 5-10. Show Camera Field of View



You can adjust the depth of the camera view by moving the slider in the Camera Depth of Field function in **Tools > Settings > 3D Display**. This does not change the depth of view the camera will have. It only provides a visual aid to determine if an object at a particular distance will be in the camera's field of view.

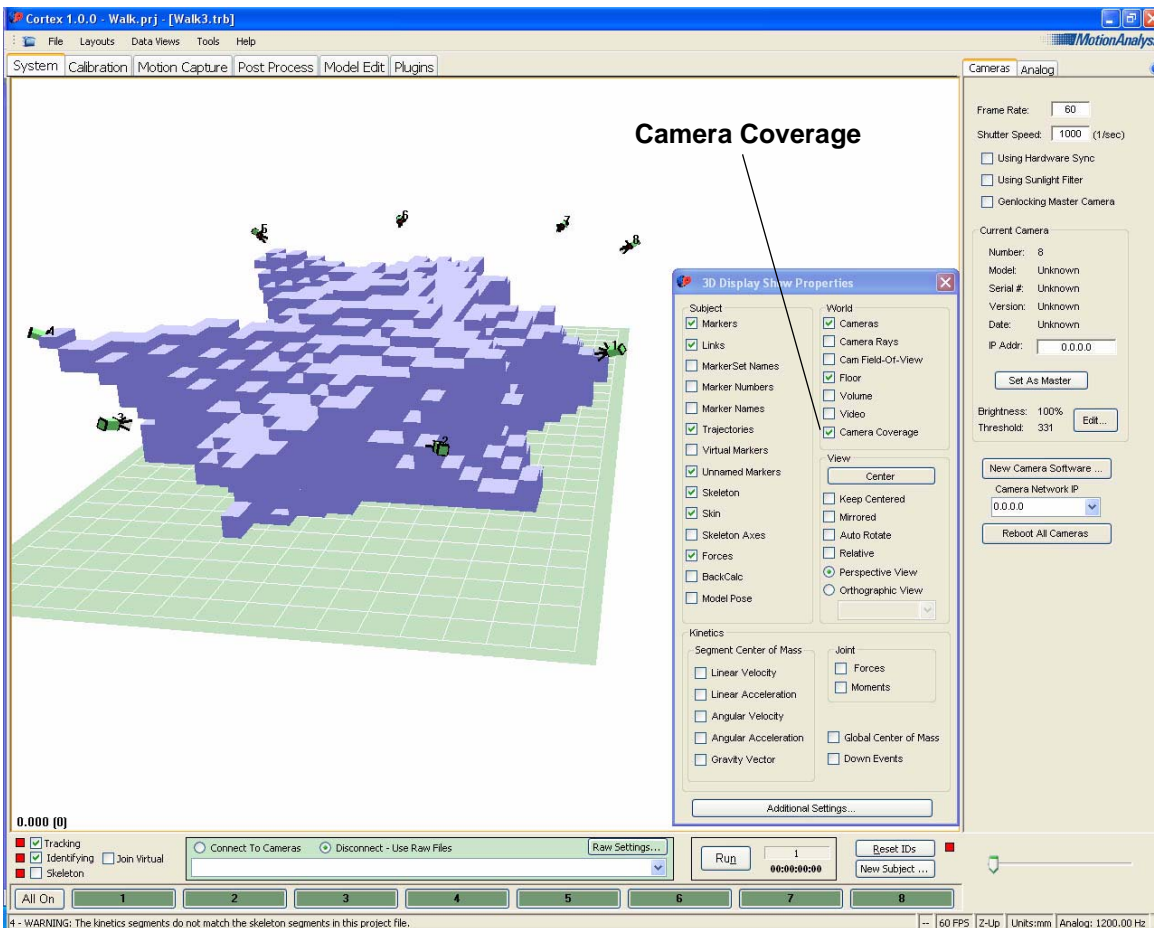
8. To see the capture volume, right-click in the 3D view, and then select the **Volume** check box from the 3D Display Show Properties window.

Figure 5-11. Show Volume



9. To See the camera coverage in the volume, select the **Camera Coverage** check box in the **3D Display Show Properties** window. See [Figure 5-12](#).

Figure 5-12. Show Camera Coverage



10. Set the shutter speed in the **Cortex** software so the markers are bright and have a good threshold setting (usually about 500).
11. Place the calibration L-frame in the center of the taped area.

Note: The cameras need at least 20 minutes to warm up before collecting calibration or trial data.

Tracking With More Than 8 Cameras

As the subject moves from one region to the next in a multiple region capture volume, **Cortex** has no problem as the subject leaves the view of some cameras while entering the view of others. The only requirement is that at least two (preferably three or four) cameras can see the subject at all times.

For additional cameras to be effective, they must be sufficiently far apart so that the rays from a given marker to the two adjacent cameras subtend a large enough angle to yield good positioning data.

**Using Many
Cameras in a Small
Volume**

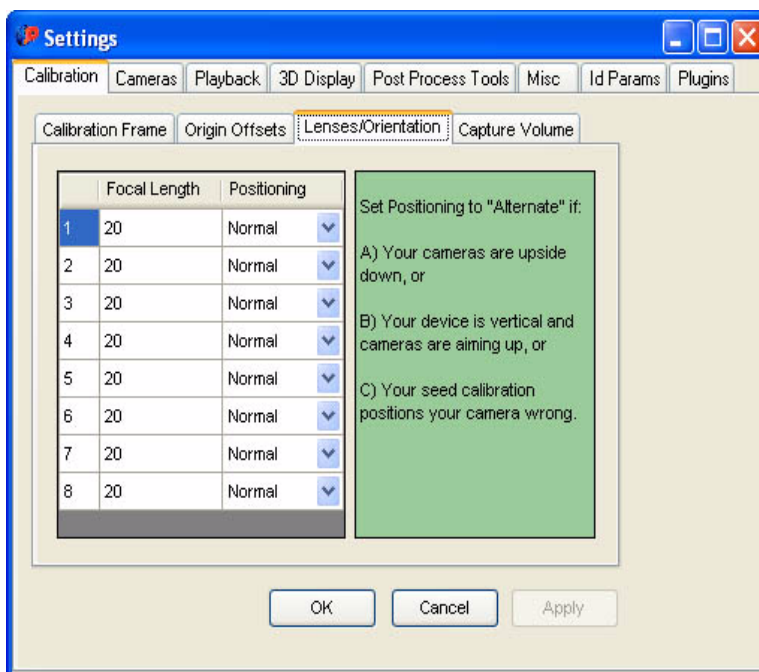
It is possible to use eight or more cameras effectively in a relatively small volume if there is sufficient height. We suggest placing half the cameras at a moderate height and the other half as high as possible. You may need to experiment to obtain the optimum camera adjustment for your lab or studio.

Adjusting Camera View for Increasing Height

If your capture volume is too high for your Eagle or Hawk cameras, you may turn the cameras on their side (just as a photographer may turn their camera on its side for increased height). Note that your camera width coverage will decrease. You may turn the cameras on their side, up to 89° without having to make any changes to the software settings. If you turn the camera 90° or greater, you will need to select the Alternate setting for the particular camera(s). This is done in the **Calibration > Calibrate** panel. Select **Details > Lenses/Orientation**, then change the setting from **Normal** to **Alternate**. These settings are also available by using the **Tools > Settings > Calibration > Lenses Orientation** tab. If the camera is hanging upside-down, you will need to use the Alternate position.

Note: It is recommended to not leave the cameras set too close to 90° (i.e. 85° to 95°) since it may appear Normal or Alternate and result in non-repeatable calibrations.

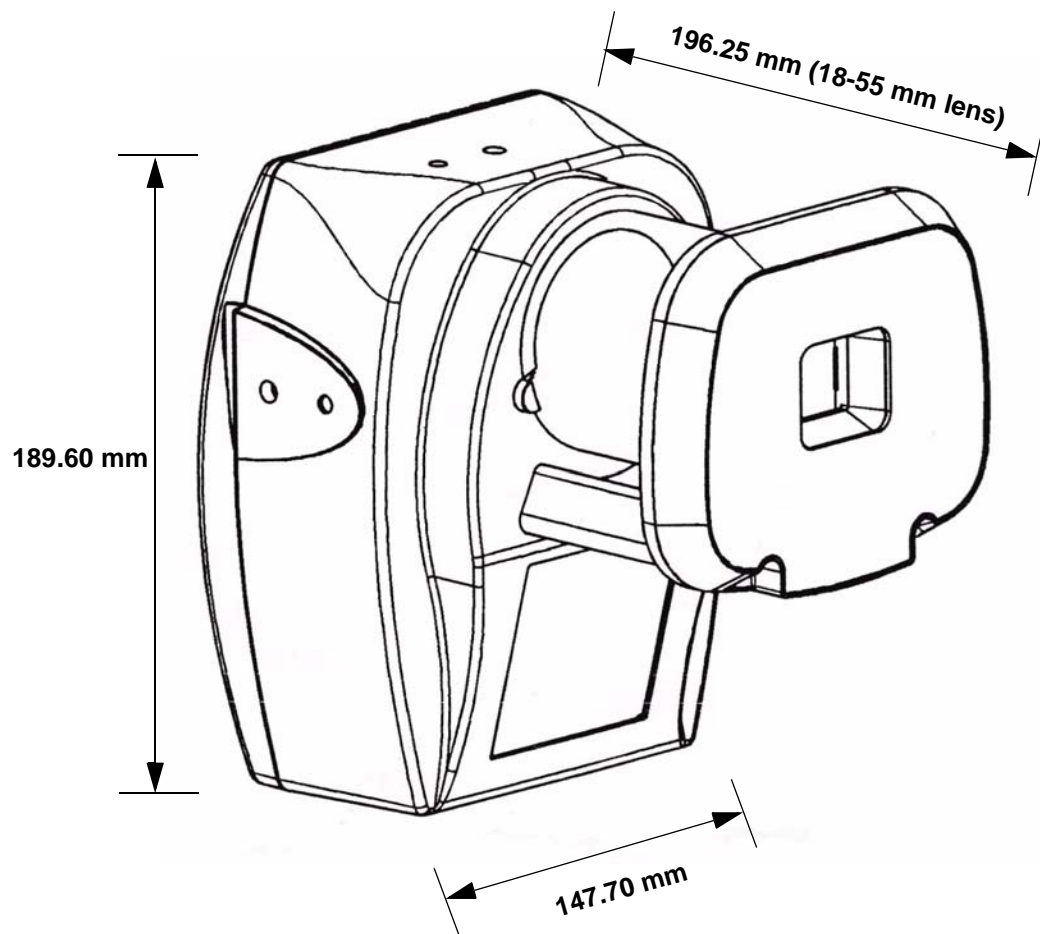
Figure 5-13. Lenses/Orientation Window



Eagle Camera Physical Dimensions

The following diagram illustrates the physical size and weight of the Eagle digital camera. The tripod mounting points are the holes used to hold the tripod bolt. There are four tripod mounting points on each camera.

Figure 5-14. Eagle Camera Physical Dimensions

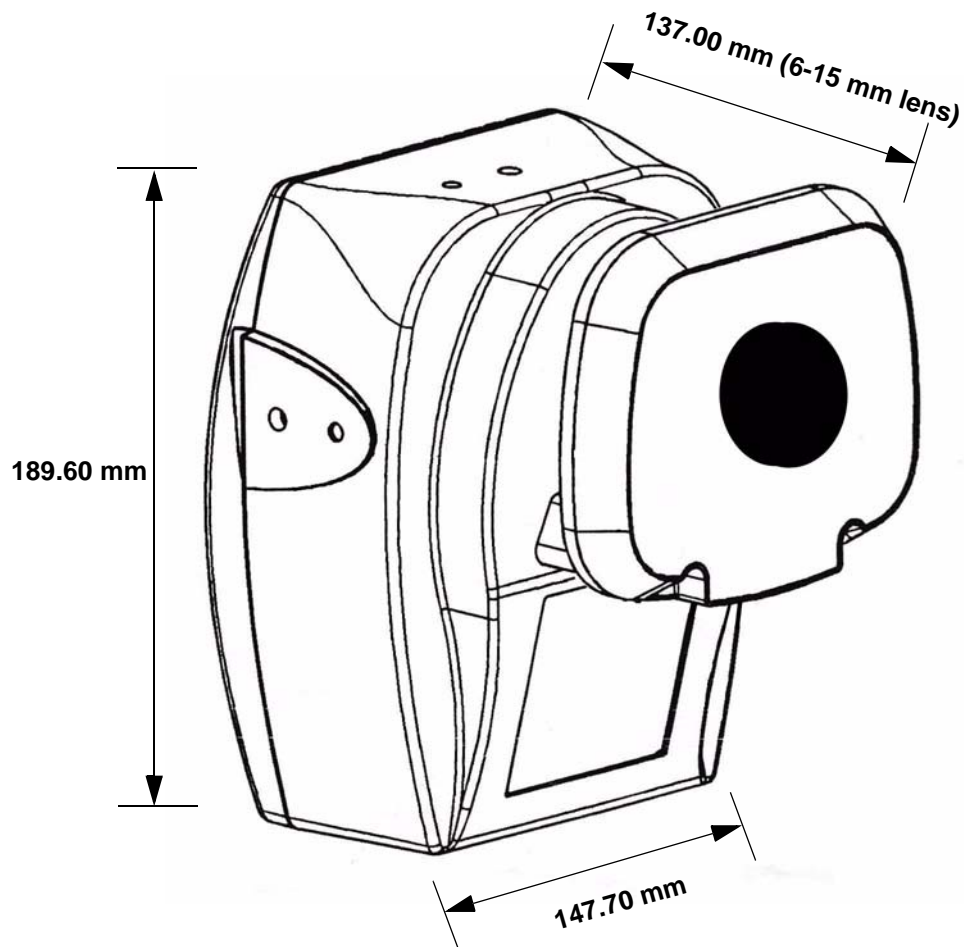


Tripod Mounting Points = 1/4 in. diameter x 20 threads/inch
Camera Weight = 2.22 kg with lens (4.90 lbs.)

Hawk Camera Physical Dimensions

The following diagram illustrates the physical size and weight of the Hawk digital camera. The tripod mounting points are the holes used to hold the tripod bolt. There are four tripod mounting points on each camera.

Figure 5-15. Hawk Camera Physical Dimensions

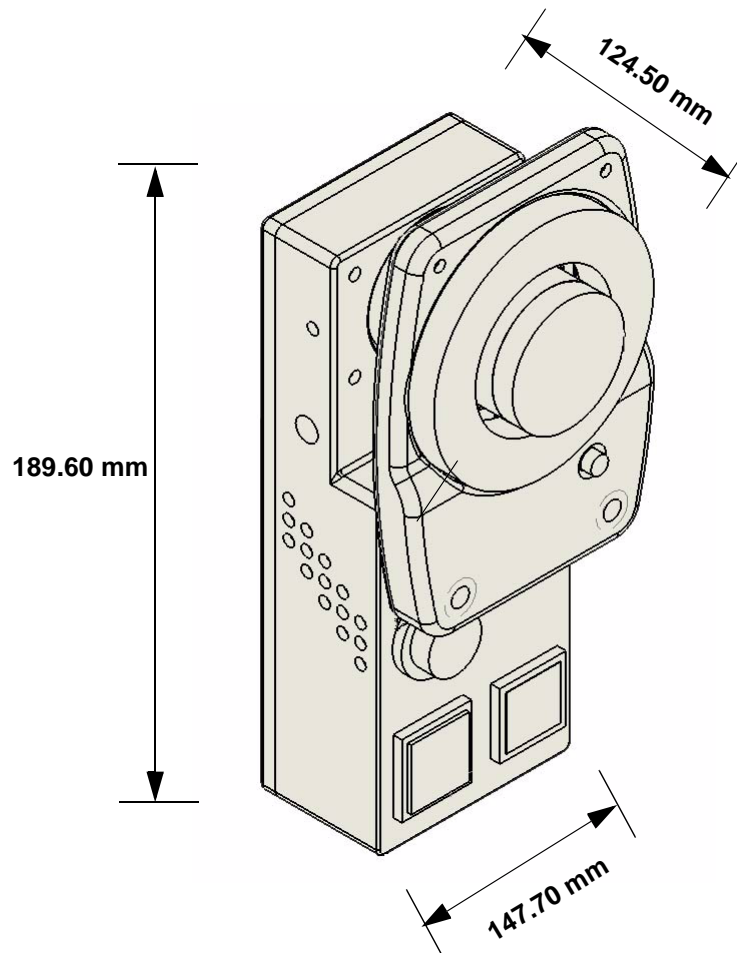


Tripod Mounting Points = 1/4 in. diameter x 20 threads/inch
Camera Weight = 2.13 kg with lens (4.70 lbs.)

Hawk-i Camera Physical Dimensions

The following diagram illustrates the physical size and weight of the Hawk-i digital camera. The tripod mounting points are the holes used to hold the tripod bolt. There is one tripod mounting point on each Hawk-i camera.

Figure 5-16. Hawk-i Camera Physical Dimensions



Tripod Mounting Points = 1/4 in. diameter x 20 threads/inch
Camera Weight = 0.77 kg with lens (1.70 lbs.)

Overview of the System Calibrating Process

System Calibration is performed in two stages: a seed and a wand calibration. A relationship must be established between real-world positions (object-coordinates) and the corresponding image-coordinates from the camera view. This is called calibrating the system. When a target is visible in two or more camera views, there is sufficient information available to track the targets in three-dimensional space.

The calibration of a given camera's view is completely dependent on the camera lens focal length and the position and orientation of the camera with respect to an arbitrary reference frame called the object-reference-frame. A change of any sort, which alters the relationship between the object-coordinates and image-coordinates, must be followed by a fresh calibration. This includes accidentally bumping a camera tripod.

The calibration process calculates eleven calibration coefficients which implicitly define the configuration of a particular view. The calibration coefficients, together with the image-coordinates of a single target, are sufficient to define the path of an optical ray from the target to the camera through the object-space. If rays from two cameras intersect in space at a specific time, they define the 3D position of a target at that time. Therefore, the tracking process is one of intersecting optical rays generated from different views of the same event. **Cortex** employs a "best fit" tracking algorithm using only good camera views.

The Calibration Coordinate System

In order to calibrate the system, you must first decide on the location of the origin and orientation of your object-reference-frame. This is determined by the calibration L-frame. All results generated by the tracking process are referred back to this reference frame.

The selection of an object-reference-frame is arbitrary. However, judicious selection is advised. In most cases, it is advisable to align one axis of the frame with the axis of gravity and another with the predominant direction of motion. Remember, that all targets will be tracked with respect to the object-reference-frame, and that the units used to locate the control points (mm, cm, inches, etc.) will be the same units used in the tracking process.

For computer animation users, a commonly used coordinate system convention is called "Y-up", with the Y axis pointing up, the +Z axis normal to the direction of motion and the +X axis oriented from the person's right side to left side. Looking at the frontal view, you would see a normal X-Y plot (Y-up, X-right) and the +Z coordinate sticks out of the screen.

For biomechanics applications, it is common to use a coordinate system with +Z up, +X in the direction of forward motion, and +Y toward the subject's left side.

The coordinate convention you use is your choice. Be sure that:

- The coordinates of the calibration L-frame are entered into your project file(s) correctly.

- The calibration L-frame is oriented correctly in the room when you collect the calibration trial. The position of the calibration L-frame determines the orientation of your calibration.

Note: The International Society of Biomechanics (ISB) has officially adopted the convention that the Y axis should point up. This has the advantage that in both 2D and 3D studies, the Y axis is up. However, many studies and software packages use the Z-up coordinate system favored by mathematicians.

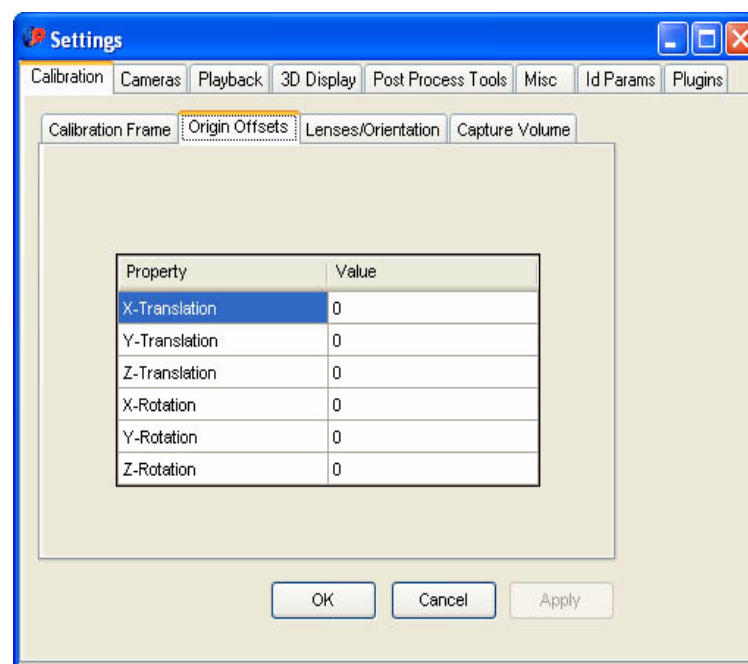
Control Points

Once a reference frame has been selected, you must provide a number of calibration markers with known locations, which can be used for control purposes; hence, these calibration markers are known as control points. The control points serve much the same purpose as the simple scale widely used for two-dimensional studies—they are, in fact, a three-dimensional yardstick representing the X, Y, and Z dimensions.

Motion Analysis offers a calibration L-frame with four retro-reflective spheres. The relative position of the spheres have been accurately measured.

Place the calibration L-frame at the origin (or at an accurately measured point) of the laboratory's test area. When placing the calibration L-frame, consider the direction of motion to be studied, position of force plates, etc. You can change the orientation of the calibration L-frame by making adjustments to the Origin Offsets table. This is located in the **Tools > Settings > Calibration > Origin Offsets** tab ([Figure 5-17](#)).

Figure 5-17. Tools > Settings > Calibration > Origin Offsets Tab

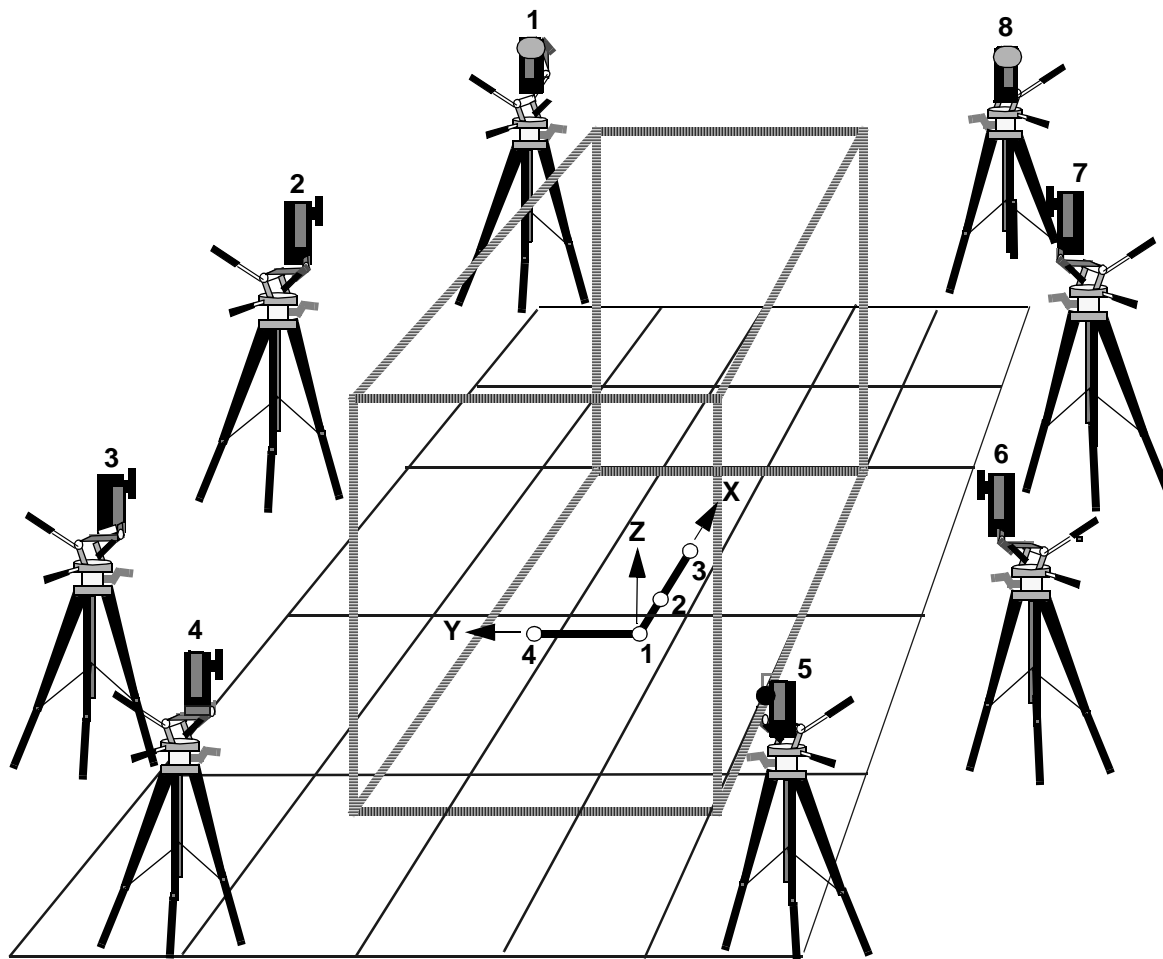


Placing the Calibration L-Frame

Mark the floor area with tape where the motion is to take place.

- Set Marker #1 of the calibration L-frame at the desired origin of the capture volume.
- If the orientation of the coordinate system is not important, the square should be rotated so as many control points as possible can be seen by all cameras.
- The calibration L-frame should be seen by at least half of the cameras to give a good calibration. The other half of the cameras can be calibrated using the wand, with an Extend Seed menu item. You may need to adjust the cameras at this point.

Figure 5-18. Placing the Calibration L-Frame In the Capture Volume (Z-Up)



Marker Sizes and Maximum Distances for Eagle/Hawk Cameras

The limiting factor in what size marker works in what volume is the distance the marker is viewable (and usable) from the camera.

"Min 3 lines" below is the minimum number of scan lines (or pixels) to allow in calculating a 2D centroid. As a rule, the more lines, the cleaner the 2D data and resulting 3D data. But once you get above 3 lines, the data is very clean and going more lines per centroid does not generally make the data any better. This setting is found in the **Tools > Settings > Cameras** tab.

Table 5-2. Marker Size and Maximum Distance for Eagle Digital Cameras

Eagle Camera Marker Size	Distance (m)	
	Min 3 Lines	Min 2 Lines
6 mm (1/4 in.)	4.2 m	7.0 m
12 mm (1/2 in.)	8.0 m	12.0 m
19 mm (3/4 in.)	10.7 m	15.0 m
25 mm (1 in.)	12.1 m	16.0 m

These are empirical tests taken from an Eagle camera with the 18-55 mm zoom lens, at a capture rate of 60 Hz, 100% brightness, and a threshold = 500.

Note: Hawk cameras will require markers that are approximately 50 to 100% larger than those listed in [Table 5-2](#).

When using 1/2-inch markers, the useful distance for VERY CLEAN data (Min. 3 lines per centroid) is about 8 meters. Going to a minimum of 2 lines per centroid, takes the usable distance to 12 meters. Going to the 3/4 inch marker gains another 3 meters (to 15 meters). A note of interest is that going to a one-inch marker does not give a big boost in distance. Other factors come into play, such as the inverse square law about light falling off as we get further from the source. Above 16 meters, the markers are "self extinguishing".

Our experience indicates that if you go beyond about 10 meters in any direction of the capture volume (length or width), it is best to have a second tier of cameras in the middle of the longer dimension. This is a big benefit for multiple person captures as it minimizes editing time needed since you get lots of good solid 3D marker points.

Troubleshooting Eagle and Hawk Camera Problems

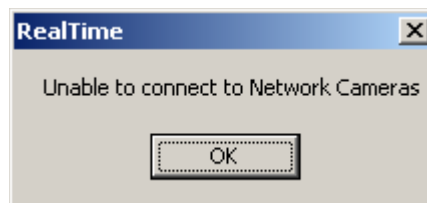
If Any Cameras Fail to Respond

Motion Analysis sets the default network address in the software to 10.1.1.199. Please note, some computers have multiple network cards installed in them. Please make sure they are labeled so there is no confusion.

If you or your IT department has changed the network address for your system or your cameras, please make note of this for reference as it will save you time in the future.

If you see the error shown in [Figure 5-19](#), there can be multiple reasons why.

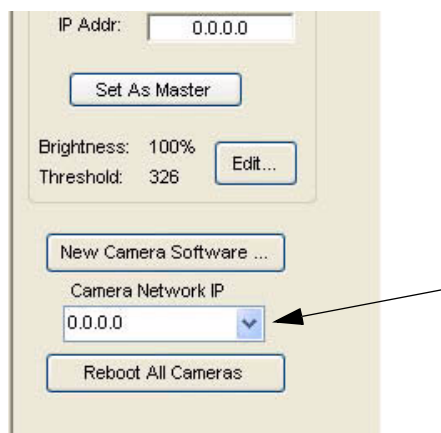
Figure 5-19. Unable to Connect to Cameras Error



The following are some steps to try and fix the problem, starting with the simplest and progressing to the more complex.

1. Under the **System > Cameras** subpanel, verify the Camera Network Address in the Camera Network Address box.

Figure 5-20. Eagle Network Address Box



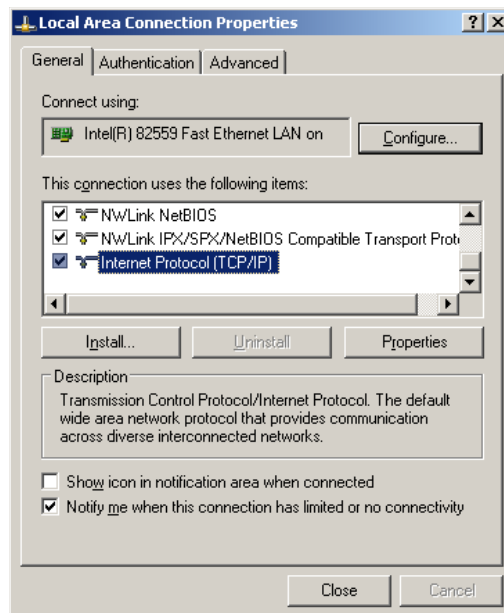
If nothing has been changed this should have a network address of 10.1.1.199. Try and connect to cameras again. If this does not fix the issue please move on to the next step.

2. Make sure that the Network cable going from the back of the computer to the EagleHub is securely installed. Motion Analysis uses the on-board network port for the Eagle Network. If you purchased a computer from another source, this may not be how your system is setup. Please take note of this when checking the connections, as it will be useful when talking to Motion Analysis Customer Support staff.
 - a. Unplug the network cable from the back of the computer and plug it back in.
 - b. Do the same for the connectors on the EagleHub. You should hear an audible "Click" when inserting it back into the jack.
3. If Windows Updates has been recently run (they may be running in the background), there may be a possibility that the Microsoft Windows Firewall was either installed or turned on. This will need to be turned off, as well as any other Firewall software installed on your motion capture computer. Because of the nature of the digital cameras, it is required that the network coming in to the computer, on a particular IP address, is open for streaming data. If your facility requires a Firewall to be installed for their network, it will need to be configured to leave the Eagle Network untouched and open.

You can turn off the Windows Firewall by going to the **Start > Settings** menu in your Windows desktop and then selecting **Control Panel > Security Center**. This is a feature in Windows XP, Service Pack 2 and later software.

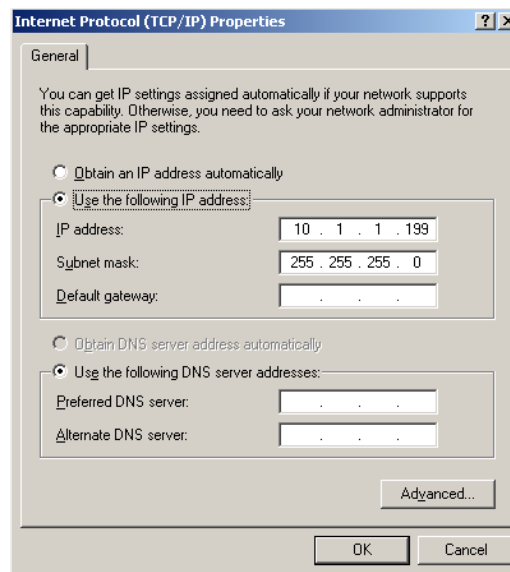
4. Double-check to make sure the Network Address that is set in **Cortex** is the same as the Network Address that is assigned to your Network Interface Card (NIC). To do this follows these steps:
 - a. From your desktop, select **Start > Settings > Control Panel > Network Connections**. You should have a “Local Area Connection” and possibly an “Eagle On-board Network” (or Hawk if so stated). There may also be “1394 IEEE Connection”, this is for Fire Wire and can be ignored.
 - b. Right-click the appropriate network connection for the Eagle Network and select **Properties** from the drop down menu.
 - c. Under the General tab, scroll down and select **Internet Protocol (TCP/IP)** and click on the **Properties** button.

Figure 5-21. Internet Protocol (TCP/IP) Properties Selection



- d. You should have “Use the following IP address” selected. If not, please make sure you have selected the proper network connection. You may close this window and return to Step 4b. The IP Address should be 10.1.1.199 and the Subnet Mask should be 255.255.255.0. If either of these is incorrect, please change them.

Figure 5-22. IP Address and Subnet Mask Address



5. If the software has still not connected to the Camera Network, use the DOS interface in Windows to Ping the cameras to verify if there isn't a hardware failure.
 - a. To Ping a camera, select **Start > Run** from the Windows desktop. In the pop up window, type in **cmd**. This will launch the command prompt.
 - b. Type in **Ping 10.1.1.201** and press **Enter**. If the request times out (it will try 4 times) try doing the next number, 10.1.1.202 and so on. Your cameras should be set to 10.1.1.201 for camera number 1 and 10.1.1.202 for camera number 2 and so on (unless changed by you or your IT personnel). If the cameras do not respond then you may need to use your Eagle Test Cable to determine the Camera Network address. The Eagle Test cable is the black cable about 1 meter in length, one end plugs into the Aux port on the back of the camera, the other end has a VGA port, a COM port and a BNC connector. Follow the steps found in Appendix A-23 through A-24; this will display the Network address of your camera.

The Ping command should return a message similar (but not exactly the same) as follows:

```
Pinging 10.1.1.201 with 32 bytes of data:

Reply from 10.1.1.201: bytes=32 time=20ms TTL=128
Reply from 10.1.1.201: bytes=32 time=20ms TTL=128
Reply from 10.1.1.201: bytes=32 time=20ms TTL=128
Reply from 10.1.1.201: bytes=32 time=20ms TTL=128

Ping Statistics for 10.1.1.201:
    Packets: Sent=4, Received: 4, Lost=0 (0% loss),
Approximate Round trip times in milliseconds:
    Minimum=0ms, Maximum=242ms, Average= 128ms
```

If you are getting a message that says:

```
Pinging 10.1.1.201 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping Statistics for 10.1.1.201:
    Packets: Sent=4, Received: 0, Lost=4 (100%
loss),
Approximate Round trip times in milliseconds:
    Minimum=0ms, Maximum=0ms, Average= 0ms
```

Then the camera is not responding to Ping requests.

6. This last step involves trying to determine if there is a camera, cable, or connection that may be causing the system to not identify the cameras. The best way to do this is to unplug all of the cameras from the EagleHub and then individually plug each camera in (do not connect any other camera cables to the hub). It does not matter which RJ45 port you connect the Ethernet switch. The same applies to the power connector. The only other connection going into the EagleHub is the cable coming from the back of the Host computer (**Cortex** tracking computer).

After you have plugged-in a camera, click-on the **Connect to Cameras** button on the Real-Time Dashboard. If this works, unplug this camera and set it off to the side or label it as good. Move on to the next camera doing the same, and so on, each time remembering that there should only be one camera plugged into the EagleHub at a time. This will help you narrow down if there is a conflict.

After running through each individual camera, it is very useful to power down the EagleHub to help clear out any stored data that may be in its memory. You may also want to take the time to write down IP address for each camera. If you run into a problem with multiple IP

Addresses being the same, this could be the problem. Each camera needs to have its own independent IP Address. No two addresses can be the same. This makes each camera unique and will help the system identify them.

Socket Error

When connecting to the cameras on the RealTime Dashboard, if you encounter a **SOCKET ERROR**, you will need to verify the following:

- that the Ethernet connector on the back of the Host Computer is working properly.
- that the Ethernet cable running from the EagleHub (or switch connected to multiple EagleHubs) to the Cortex Host Computer is connected.

Function Not Found In Library Error

If you get the message **ERROR: Function not found in library**, you have an older version of a library that is not working or is needed. If you do not have the A-D option, check the **C:\Winnt\System32** directory and rename the file **nidaq32.dll** to be **nidaq32.dll.old**. Then close and relaunch **Cortex**. This applies for Windows XP operating systems. If you are using a different operating system, you will need to do a search for the file **nidaq32.dll**.

Relationship Between Capture Volume and Marker Size

Listed in [Figure 5-3](#) are Optimal (Highest Accuracy), Large Volume, and Extended Volume capture areas for the 6, 8, and 10 camera systems. Larger volumes require more cameras. Different shaped capture areas are also possible.

At the extremes, volumes will vary with ceiling heights and can vary with optical conditions including external lighting.

Table 5-3. Guidelines for Selecting Marker Size

	Marker Size	Normal Capture Volume	
		meters	# Cameras
Highest Accuracy	Eagle = 3/8-inch (9mm)	2.5 x 2.5	6
	Hawk = 1/2-inch (12.5mm)	2.5 x 3.5	8
Large Volume	Eagle = 1/2-inch (12.5mm)	3.5 x 3.5	6
	Hawk = 3/4-inch (19mm)	3.5 x 6	8
Extended Volume	Eagle = 3/4-inch (9mm)	5 x 5	6
	Hawk = 1-inch (12.5mm)	5 x 8	8
1/8" and 1/4" markers are also available for smaller volumes such as face, hand, or foot capture volumes.			

Topic	Page
Getting Acquainted With the User Interface	6-1
PRJ Files	6-30
Viewing Sample Data	6-31
Digital/Reference Video Option (EVaDV Software)	6-40
Real Time Dashboard	6-40
Post Process Dashboard	6-50
Post Process Dashboard	6-50
Post Process Tool Strip	6-50
Zooming, Rotating, and Translating in the 3D View	6-51
Selecting Markers, Virtual Markers, Linkages, and Segments	6-52
Time Code	6-53

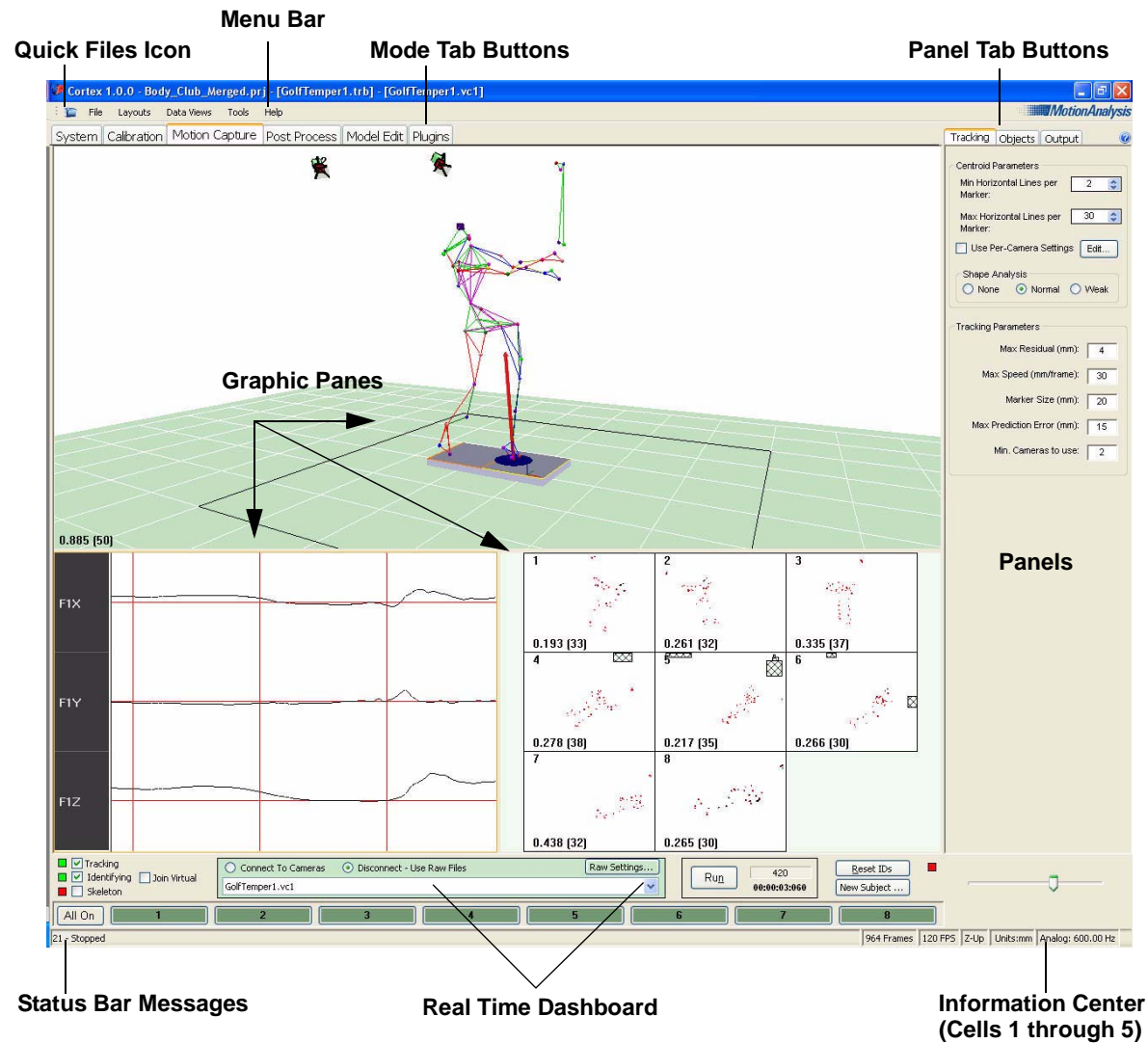
Getting Acquainted With the User Interface

Before using **Cortex** it is necessary to become familiar with the interface and the names of the tools and controls that will be used throughout this manual. The major components are as follows:

1. The Graphics Panes in the center of the screen
2. The Menu Bar in the top-left side
3. The Directory List below the Menu Bar
4. The Mode Buttons along the top left of the screen
5. The Panel Buttons along the top right of the screen
6. The Panels on the right side of the screen
7. The Real Time Dashboard along the bottom of the screen
8. The Post Processing Dashboard replaces the Real Time Dashboard. while the program is in Post Processing mode
9. The Status Bar Messages in the lower-left corner
10. The Information Center in the lower-right corner

An image of the interface is shown in [Figure 6-1](#). Note that multiple 3D views can be rendered simultaneously.

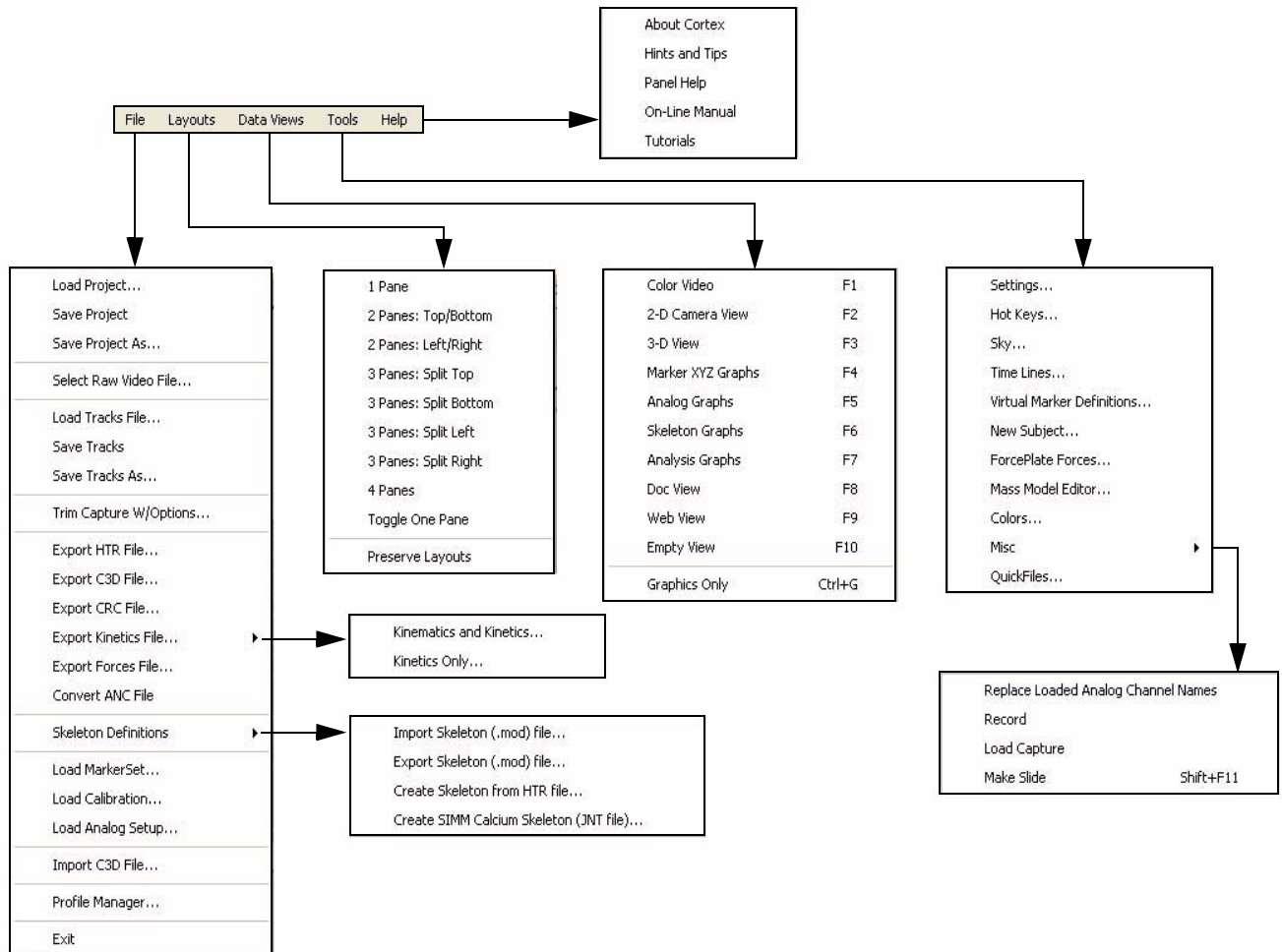
Figure 6-1. Cortex Interface in Real Time Mode



Menu Bar

The Menu Bar selects the primary items for **Cortex** functionality. These include file management, layout control, data views, tools, and help.

Figure 6-2. Menu Bar



File Menu

Load Project...

Loads a project file (.prj) from the current working directory.

Save Project

Saves a project file (.prj) to the current working directory.

Save Project As...

Provides a method to save the current project (PRJ) file under a different name.

Select Raw Video File (Live)...

Loads the VC (Raw Video) files from a capture session.

Load Tracks...

Loads the TRB or TRC (Tracks) files from a capture session.

Save Tracks

Allows the user to save the current Tracks (TRB or TRC) file with the current filename.

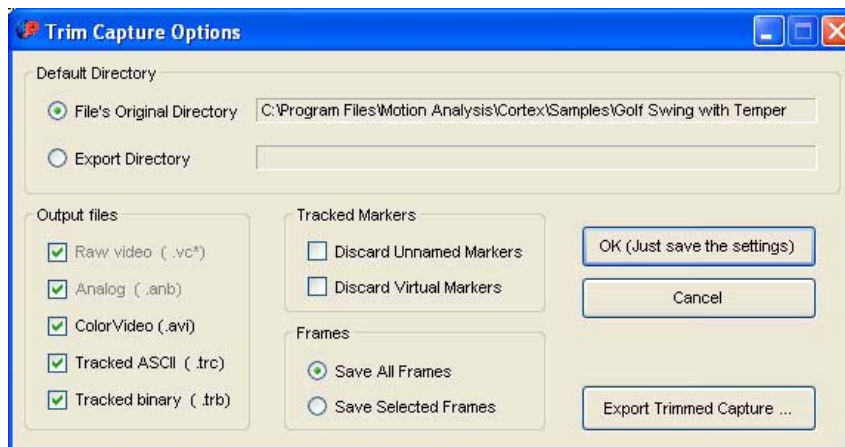
Save Tracks As...

Provides a method to save the current Tracks (TRB or TRC) file under a different name. This function will only allow you to change the file name, not the file type. To change the file type, you must use the **Export.xxx File...** menu item.

Trim Capture W/Options...

Provides frame and marker options when saving Tracks (TRB or TRC) files. The **Trim Capture** feature allows you to specify which directories the new files will be saved to, as well as frame and marker management options. Refer to [Figure 6-3](#). The software remembers two folder names, the current folder, and the export folder.

Figure 6-3. Trim Capture W/Options Interface

**Export HTR File...**

Exports a file in HTR format, which is commonly used in the animation software packages.

Export C3D File...

Exports marker positions and analog data in an open sourced file format viewable by many different software packages.

Export CRC File...

CRC (Centroid Row Column) data are the 2D data points in ASCII text format. Usable by advanced users who want to reconstruct 3D positions of markers using their own software, in post process mode only.

Export Kinetics File...**Kinematics and Kinetics...**

Exports Kinematics (joint angles) and Kinetics (forces and moments) data into a single .kin file, which is a tab-delimited ASCII file easily viewed in Microsoft Excel™ spreadsheets.

Kinetics Only...

Exports Kinetics (forces and moments) data into a single .kin file, which is a tab-delimited ASCII file easily viewed in Microsoft Excel™ spreadsheets.

Export Forces File...

Exports ASCII files containing forceplate data. This uses your current forcepla.cal file and converts the raw forceplate data into calibrated forces. The units used are Newtons and Newton-Meters and each line in the file equates to one analog sample.

Convert ANC File...

Converts binary formatted Analog data (.anb) into an ASCII viewable format (.anc).

Skeleton Definitions

Import Skeleton (.mod) File... provides a method for bringing a new MOD file into the current PRJ file. MOD files are used for Deep Solver applications. The name of the MOD file will match the marker set name as defined in the project.

Export Skeleton (.mod) File... provides a method for saving a MOD file for use in a different PRJ file.

Create Skeleton from HTR File... brings in the skeleton definitions from the animation package.

Create SIMM Calcium Skeleton (JNT File)... calculates and stores a JNT file which is scaled to the subject's bone sizes. This JNT information is also stored in the current project file when saved to memory.

Load Marker Set...

Can be used to load a previous marker set into a new or newly calibrated project.

Load Calibration...

Can be used to load a new set of calibration information into a project.

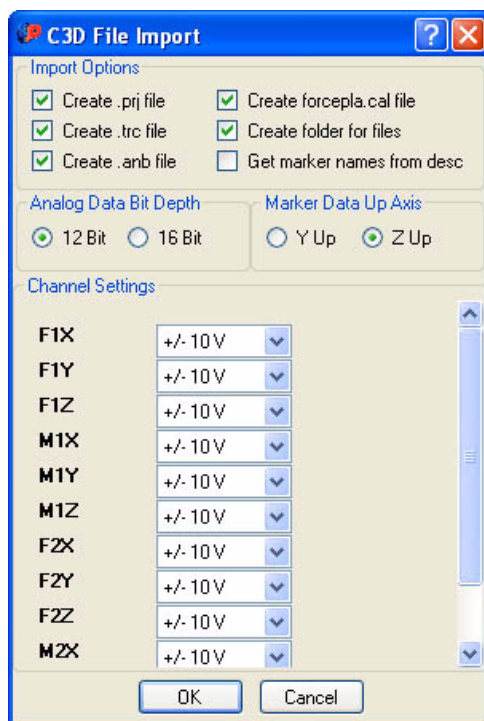
Load Analog Setup...

Can be used to load a new set of analog setup information (forceplates, EMG, and others), into the current project.

Import C3D File...

C3D is a file format used by some application for storing motion and analog data. The C3D File Import Dialog allows data stored in this format to be imported into Cortex. Some information about the capture configuration is needed to convert C3D into the corresponding MAC file types.

Figure 6-4. Import C3D File Interface



Create .prj file

If this box is checked, a prj file will be generated for the c3d project being imported. If this box is not checked, the import will load these settings without saving them to a file.

Create .trc file

If this box is checked, a trc file will be generated, containing the motion data from the c3d project being imported. If this box is not checked, the import will load the motion data without saving it to a file.

Create .anb file

If this box is checked, an anb file will be generated, containing the analog data from the c3d project being imported. If this box is not checked, the import will load the analog data without saving it to a file.

Create forcepla.cal file

If this box is checked, a forcepla.cal file will be generated, containing the force plate setup for the c3d project being imported. If this box is not checked, the import will load these settings without saving them to a file.

Create folder for files

If this box is checked, a folder will be created in the directory containing the c3d file. All generated files will be saved into this directory. The folder will be named <c3d project name>_Cortex.

Get marker names from desc

If this box is checked, the marker names in the c3d project are taken from the marker description, instead of the marker name. The c3d file format restricts marker names to 4 characters, so when exporting from an application without these restrictions (Cortex), these names can be preserved in the marker description field of the c3d file.

Analog Data Bit Depth

This setting specifies the bit-depth of the analog to digital converter used in capturing the original analog data.

Marker Data Up Axis

This setting specifies the up axis of the global coordinate system that was used when the motion data was captured in the original c3d project.

Channel Settings

These settings are used to specify the voltage ranges of the analog channels that captured the analog data in the original c3d project.

Profile Manager...

The Profile Manager opens a dialog which manages the customized settings for individual users. These customized settings include the following:

- Hotkeys
- Colors
- Layouts and Data Views
- Post Process Tool Strip

The Profile Manager dialog has three main components:

- Current profile set for manipulation (top)
- List of all available profiles (left side)
- Profile manipulation buttons (right side)

Figure 6-5. Profile Manager



Create New

A prompt to enter a profile name will be displayed. A profile with that name and the default settings will be created. This will be set as the current profile.

Load Selected

This will load whatever profile is selected from the profile list and make it the current profile.

Delete Selected

This will delete the profile selected in the profile list. Deleting the current profile is not allowed.

Save Current

This will save the current profile. Saving is also done automatically when the application closes.

Save Current As...

A prompt to enter a profile name will be displayed. A profile with that name and the current settings will be created and made the current profile. Note: the current settings are not necessarily the same as the settings for the current profile. If any settings changes have been made prior to the Save As, these will be reflected in the new profile and not the current profile.

Import Selected...

This launches the profile import dialog. This allows portions of another profile to be loaded for use with the current profile.

Exit

Allows you to exit the **Cortex** software. Make sure you have saved the current project and tracks files.

Layouts Menu

The items in this menu are generally self explanatory.

Data Views Menu

The **Cortex** interface can accommodate up to four simultaneously open data views and can be resized with the mouse to fit the panes however you desire. You can view combinations of six different graphical panes, which include:

Color Video F1

Displays the live action in a capture volume or a replay of an AVI file. You will need to have a DV (Digital Video) camera connected to the Host computer's 1394 Firewire port to see the live video in this window.

2-D Display F2

Displays a 2D image of the markers and their centroids.

3-D Display F3

Displays a moving 3D stick figure showing named markers, virtual markers, linkages and/or a skeleton.

Marker XYZ Graphs F4

Displays graphs depicting the marker's positions in each frame.

Analog Display F5

Displays analog data graphs representing the force plate's output.

Skeleton Graphs F6

Displays the following data graphs for selected skeletal segments:

- Kinematics
- Forces
- Moments
- Linear Movement
- Angular Movement
- Solver Stats

Analysis Graphs F7

Calculates and displays angles between markers, distances between markers, and position, velocity, and acceleration of a marker, or groups of selected markers. Results can be saved as ASCII Time Series (.ts) files. For more information, refer to [Appendix K, Analysis](#).

Doc View

Provides an interface to open and edit most text based files and display them within the **Cortex** software.

Web View

Provides an interface to open, display, and edit HTML files within the **Cortex** software.

Empty View

Leaves a blank space for the selected pane.

Graphics Only (Ctrl+G)

Hides the side panels and Post Process/Realtime Dashboard to maximize the graphic panes.

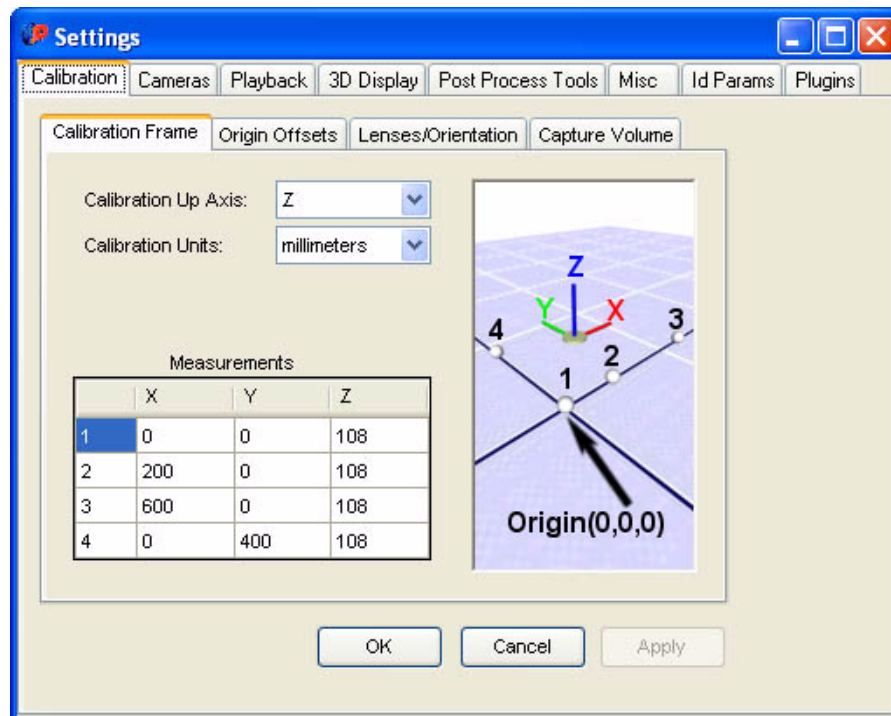
Tools Menu

The Tools menu gives access to various functions within the **Cortex** software. Many of these functions are also available in other sections of the user interface.

Tools > Settings

Calibration Tab

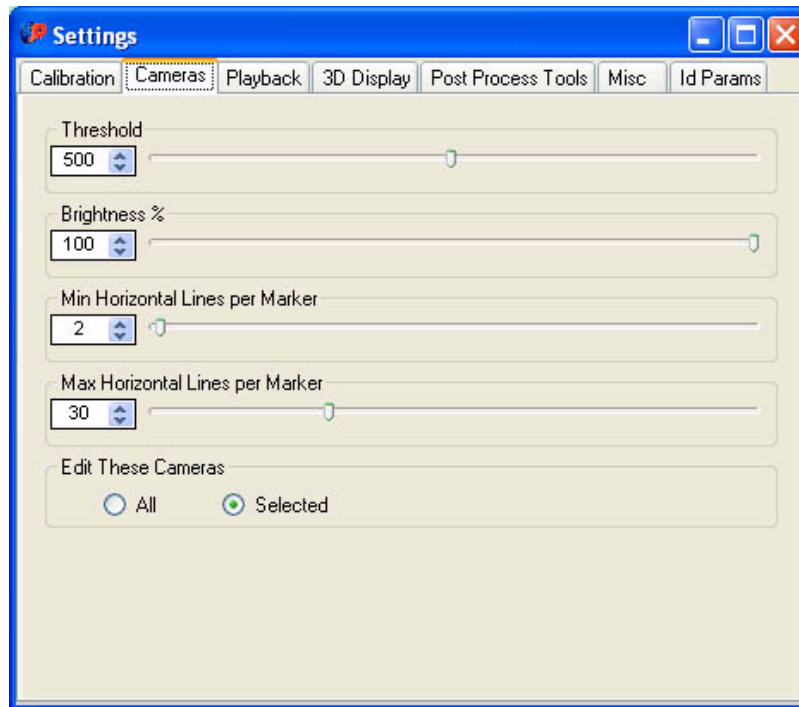
Figure 6-6. Settings > Calibration Tab



The Calibration Setup defines the parameters for the system calibration (e.g. capture volume up-axis, calibration units, etc.). For complete information, refer to [“Details... Button \(Calibration Settings Window Tabs\)”](#) on page 8-4.

Cameras Tab

Figure 6-7. Settings > Cameras Tab

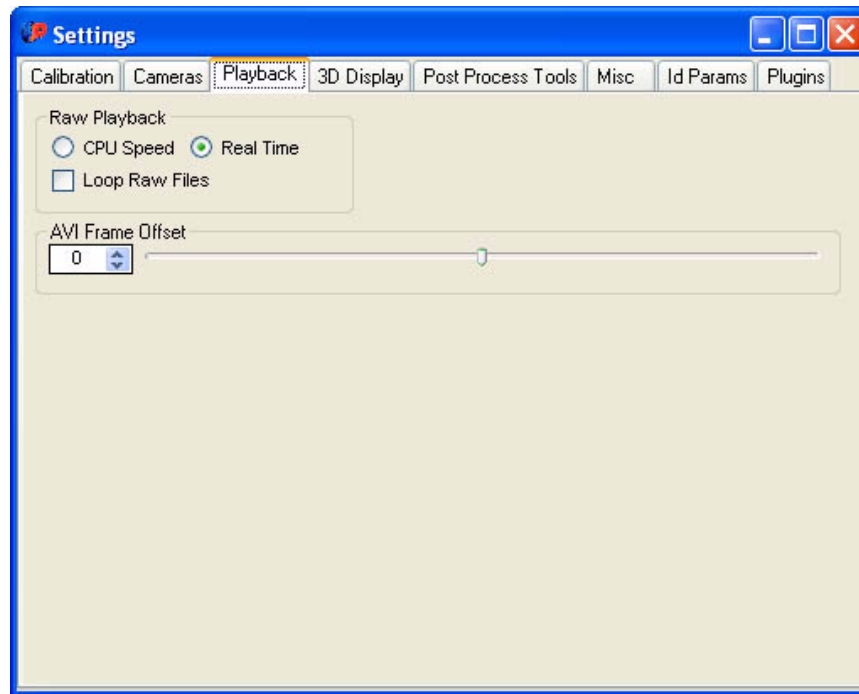


This tab provides the adjustment sliders for Threshold (used to block out excessive noise in the 2D camera view) and Brightness (ringlight) settings, Min Horizontal Lines per Marker, and Max Horizontal Lines per Marker.

For more information on Thresholds, refer to [“Adjusting Thresholds” on page 7-9](#).

Playback Tab

Figure 6-8. Settings > Playback Tab



The Playback Tab sets the speed at which the raw video (.vc) files are displayed in the 3D view and run through the Cortex software. There is also a function for adjusting the reference video (AVI) files.

Raw Playback

CPU Speed runs the file as fast as the CPU will allow. The Real Time setting will run the data at normal speed.

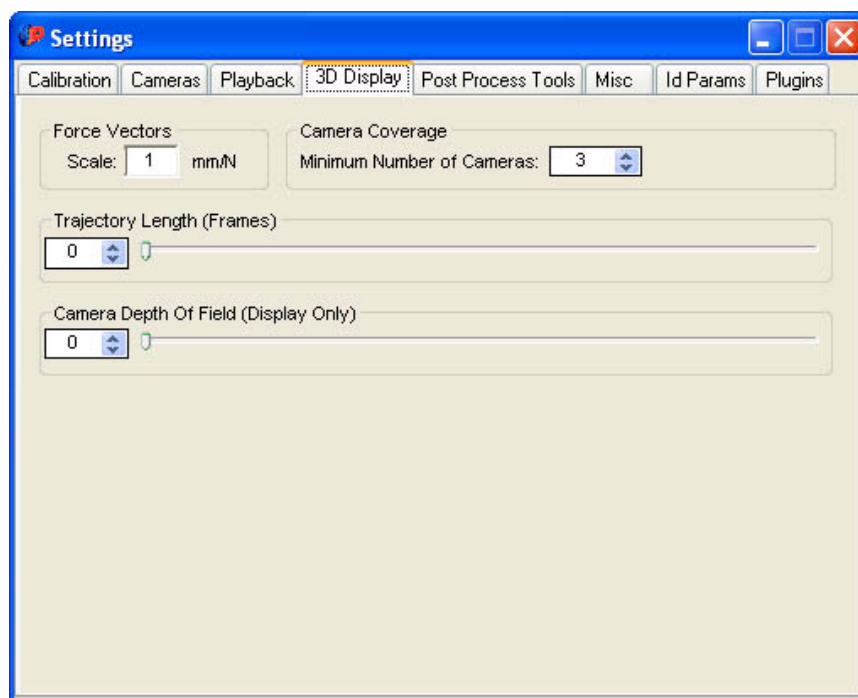
Another option under Raw Playback is the Loop Raw Files check-box. This will run the raw video file in an endless loop, until the stop button is selected. Note that this check-box is inactive when in the CPU Speed mode.

AVI Frame Offset

If a delay occurs between the reference video (AVI) files and the raw video or tracks files, this adjustable slider is used for correction. Offsets are limited to 100 frames in both the + and – temporal direction.

3D Display Tab

Figure 6-9. Settings > 3D Display Tab



Force Vectors

Sets the scale for the force vectors as shown in the 3D view. Scale values can range from 0 to ± 100 mm/N. A negative scale setting will show the forces below the forceplate.

Camera Coverage

Sets the number of cameras used in the Camera Coverage function. A small number in this setting results in a large camera coverage display (shown as blocks). A larger number (limited to the number of cameras in the system) will show a smaller coverage volume.

Trajectory Length

Shows the path history of the selected marker in the post process (only) 3D view.

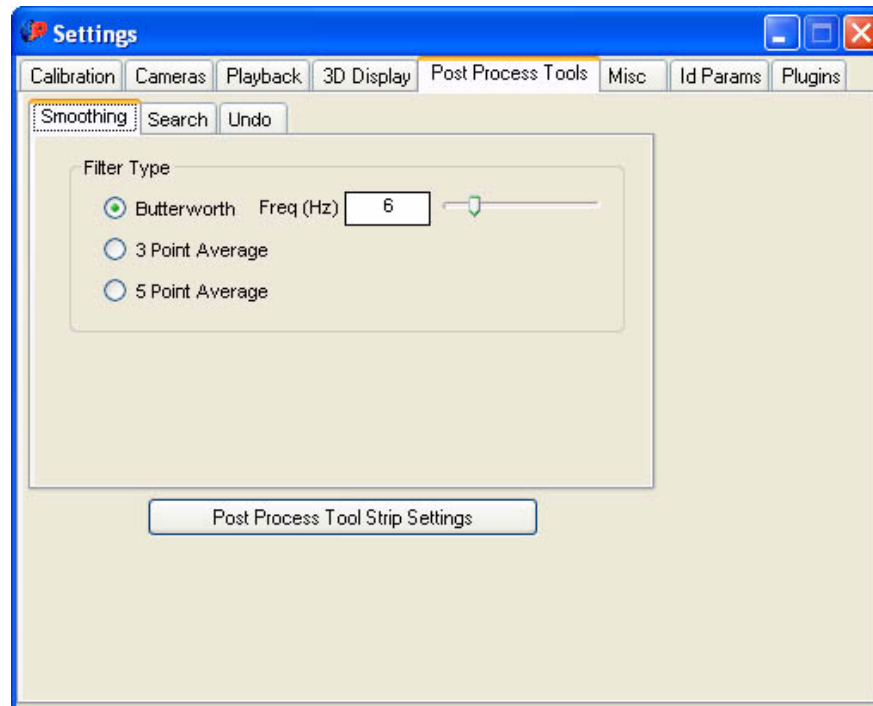
Camera Depth of Field

Used for display purposes, this function shows a representation of the selected camera's view as a cone. The Cam Field-Of-View check-box in the 3D Display Show Properties (right-click 3D view, select **Show**) must be active. The width of the cone is dependent on the camera lens focal length setting.

Post Process Tools Tab

Smoothing Tab

Figure 6-10. Settings > Post Process Tools > Smoothing Tab

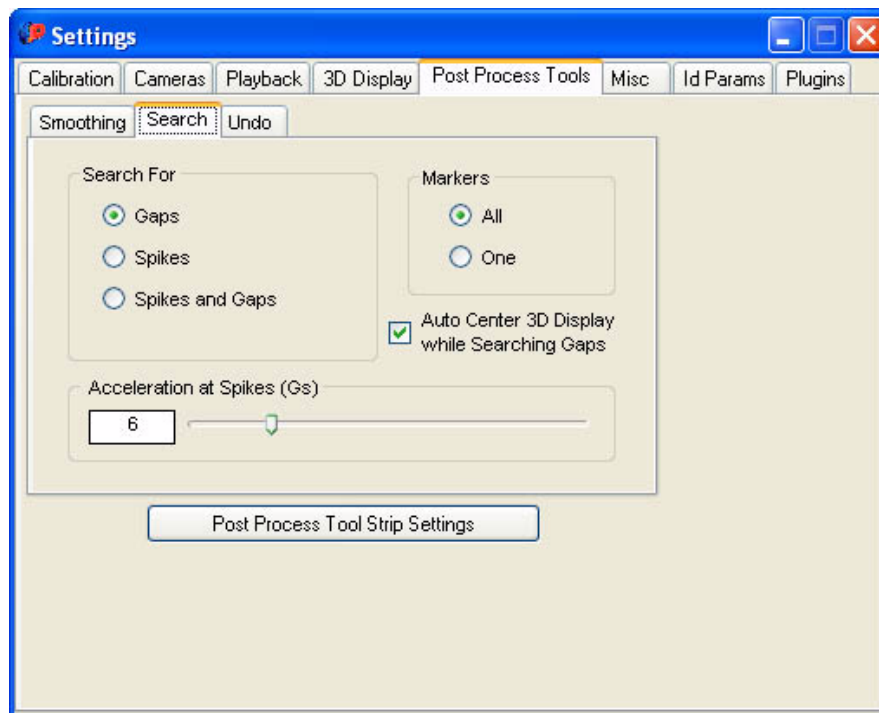


The Smoothing tab allows the user to select which automated smoothing filter to use for post processing. For more information on these filters, reference [“Filters” on page 10-5](#).

This tab also allow the user to select the Post Process Tool Strip Options. For information on the Post Process Tool Strip, reference [“Post Process Tool Strip” on page 6-50](#).

Search Tab

Figure 6-11. Settings > Post Process Tools > Search Tab

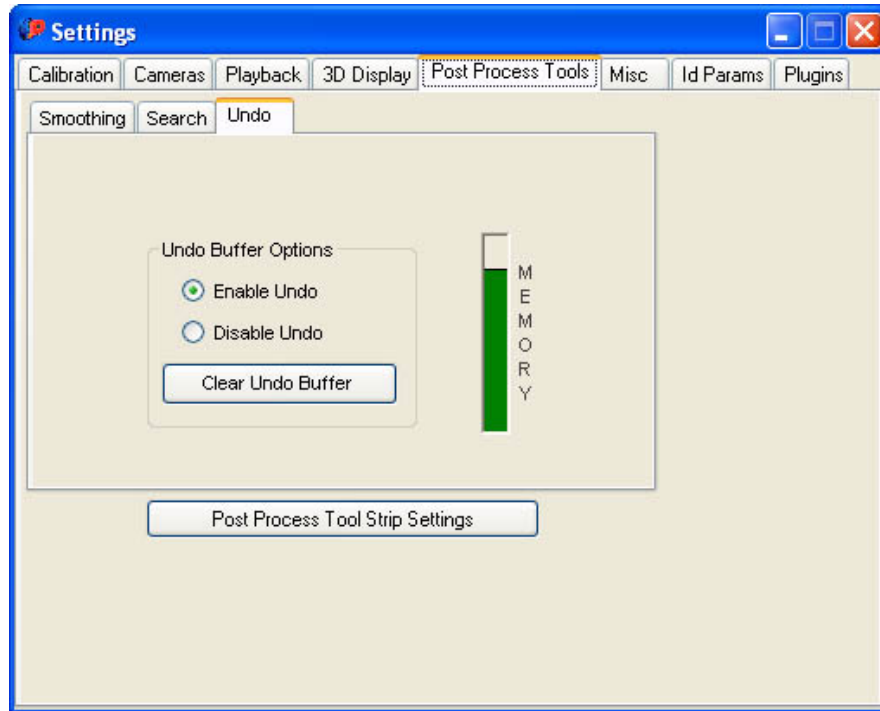


This tab sets the search parameters for finding specific discontinuing and inconsistent post process data.

User's can set the search for gaps and/or spikes within the data for all and individually selected markers. There is also an adjustment slider for the acceleration spikes.

Undo Tab

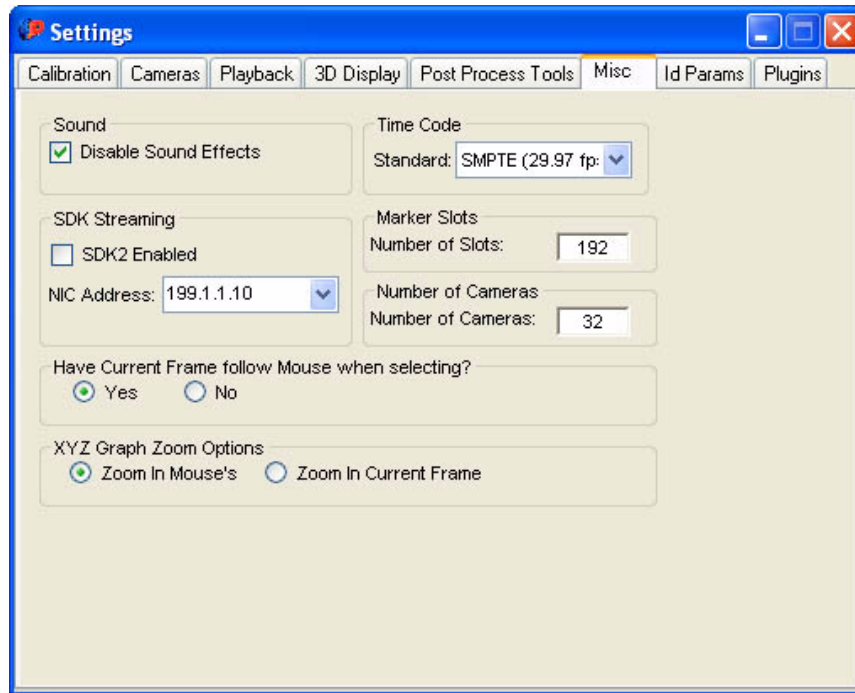
Figure 6-12. Settings > Post Process Tools > Undo Tab



This tab allow the user to turn on and off the Undo function. The Undo function can use high levels of memory and this provides the option to turn it off. The Undo Buffer can also be cleared out to lower memory usage by selecting the **Clear Undo Buffer** button.

Misc Tab

Figure 6-13. Settings > Misc Tab



Disable Sound Effects

This turns the sound features of **Cortex** on and off.

SDK Streaming Options

This option streams all motion capture and post process playing data to the NIC address specified. This must be a NIC (Ethernet) address of the **Cortex** host computer. If there are multiple NIC cards in the host computer, you must indicate which card will be used to stream the SDK2.

Time Code

This function sets the type of Time Code used with the system (if used).

Marker Slots

This function depicts the number of marker slots the software will use. **Cortex** allocates memory for the host computer based on this number, and will variably assign memory according to the number of marker slots used. The maximum number of marker slots that **Cortex** will allow is 1500. The default number of marker slots is set to 192, which uses 512 MBytes of memory. From 193 to 500 markers, the amount of memory used is 750 MBytes. From 501 to 1500 markers, the amount of memory used is 1000 MBytes.

Note: Upon changing the number of marker slots, you must quit and relaunch **Cortex**.

Number of Cameras

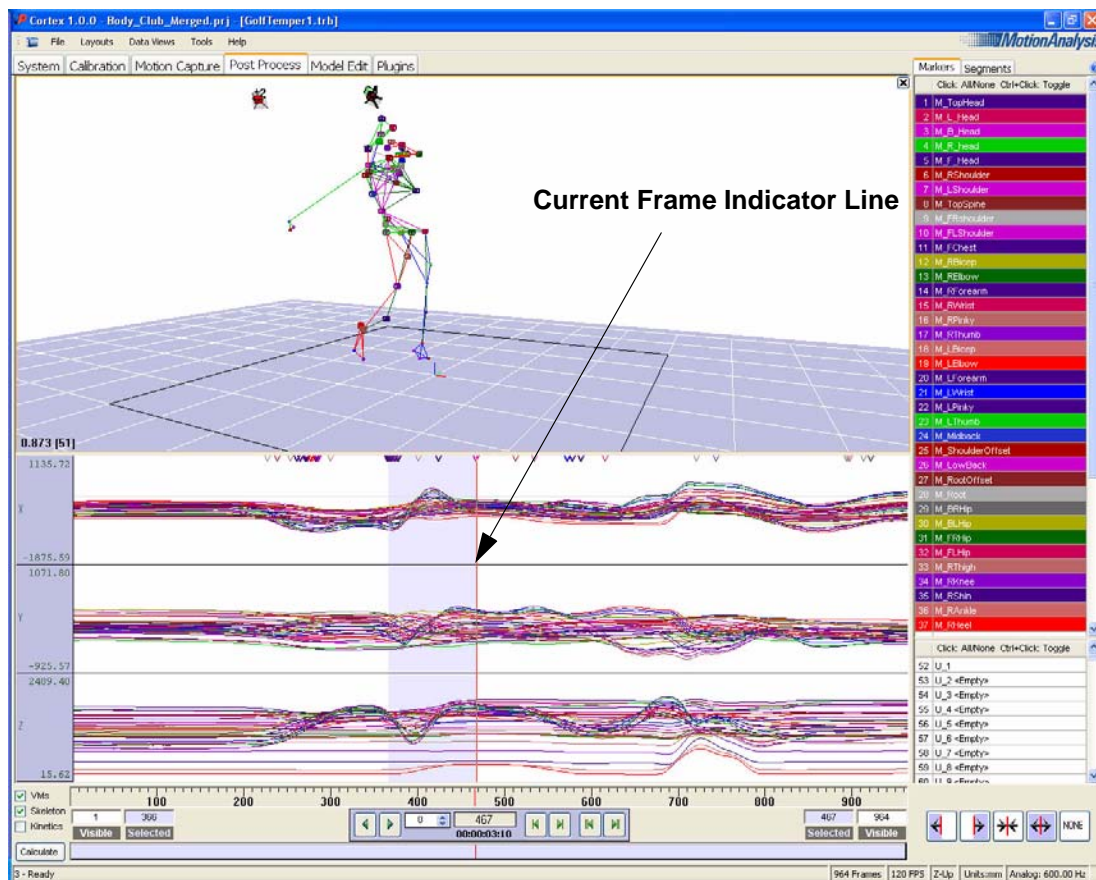
This function sets the current number of cameras used with the **Cortex** software. This number determines how much memory is allocated for the host computer and scales the software accordingly. The default number is 32 cameras.

Note: You can reduce the memory usage by lowering the number of cameras in this box. Changing from 32 cameras to 8 cameras reduces memory usage by about 200 Mbytes (7.5 to 8 Mbytes per camera). If your system is slow or limited by memory usage, reduce this number to the actual number of cameras in the system for optimal performance.

Have Current Frame follow Mouse when selecting?

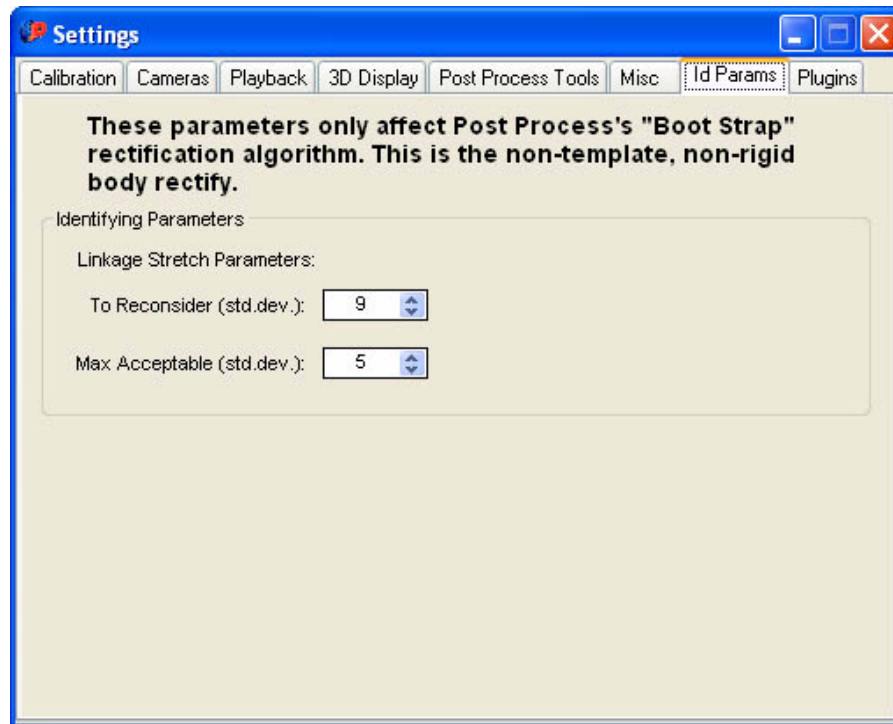
With this option selected, the current frame indicator line will follow the mouse movement across the graph when selecting a specific frame range. The current frame indicator line is the red, vertical line in XYZ, Skeleton, Analysis, or Analog graphs (see [Figure 6-14](#)).

Figure 6-14. Current Frame Indicator Line



ID Params Tab

Figure 6-15. Settings > ID Params Tab

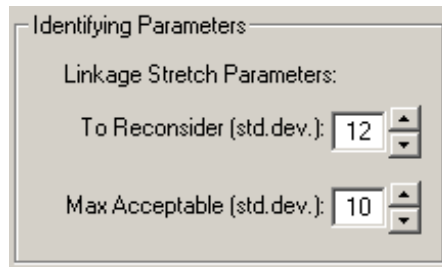


This is used to automatically identify markers based on a Model Pose that you create when you make a template. The result is that if you use the same marker set repeatedly, you will not have to ID the new person each time the marker set is used. This is the same function as using the **New Subject...** button that is found on the Real Time dashboard. For complete information, refer to [“New Subject Button” on page 6-43](#).

To Reconsider

If a link stretches more than the set amount, the path is snipped into two paths where the link stretches too much. This happens if markers come together and pull apart and the identity is not correct when they pull apart. The software may not see it right away, but after a few frames, the linkage for the wrongly named markers get too long and the path is cut. The bigger the number, the more the link is allowed to stretch before it is cut. Smaller means fewer errors, more cuts. Larger means more stretching is allowed before the cuts. It is measured in multiples of the standard deviation of the linkage length to make it accommodate linkages that normally change a lot (head to shoulder) and linkages that do not change much (elbow to wrist). Also used in Real Time streaming (Run) mode.

Figure 6-16. Linkage Stretch Parameters

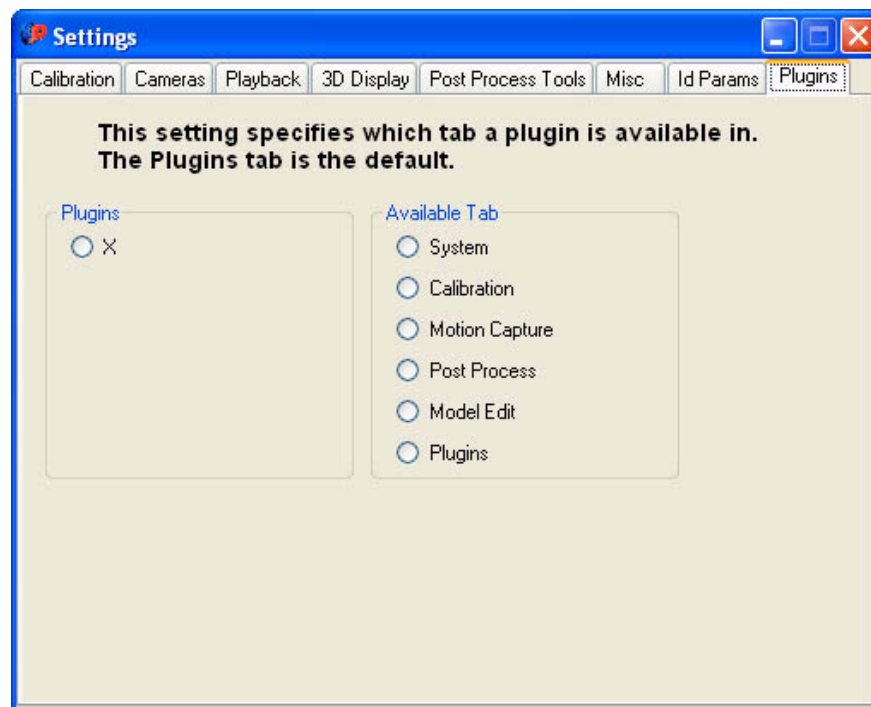


Max Acceptable

If there is a missing marker and there is an Un-named marker within this distance of known linkages, the Un-named marker is accepted as a Named marker. Also used in Real Time streaming (Run) mode.

Plugins Tab

Figure 6-17. Settings > Plugins Tab

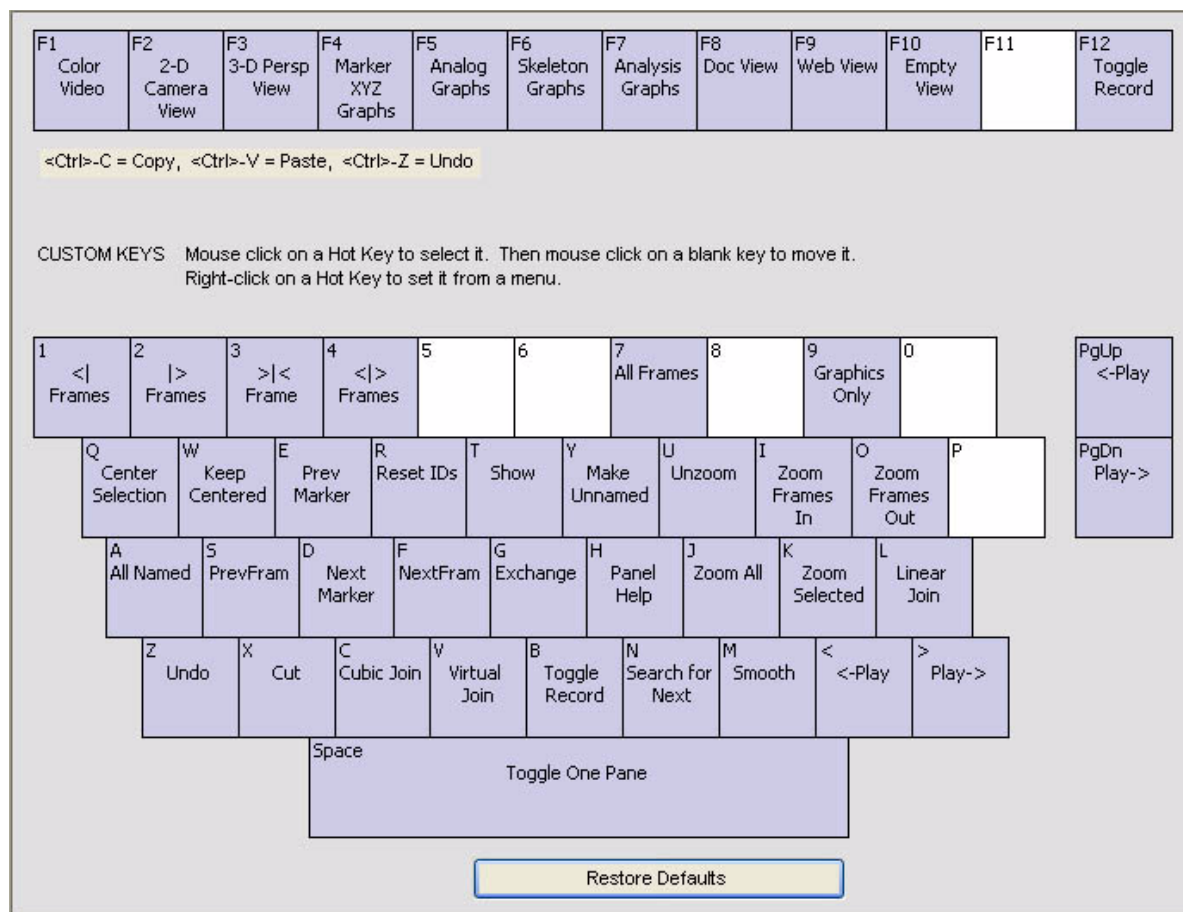


This provides a custom function to set where the Plugins tab will reside within each of the main tabs (System tab, Calibration tab, Motion Capture tab, etc.). The default tab for the X panel is in Plugins.

Hot Keys

Hot Keys are short-cuts which integrate complex procedures into single key-strokes. There are default settings initially, as shown in [Figure 6-18 on page 6-22](#), and users can also create their own custom Hot Key functions.

Figure 6-18. Default Hot Keys Setup



Sky

For Sky information and functions, please refer to [Chapter 14, Sky Scripting Interface](#).

Time Lines

Shows the time line of the data for each marker, indicating any breaks in the stream of data. More information can be found in ["Time Lines" on page 10-26](#).

Virtual Marker Definitions

This sets the definition markers that are used to support a particular virtual marker. For complete information regarding Virtual Markers, refer to [“Virtual Markers” on page 11-17](#).

New Subject

Please refer to [“New Subject Button” on page 6-43](#).

Forceplate Forces

Selecting this feature will display the forceplate measurements in numerical values. This works when you are live and connected to the cameras, or when you are simulating Real Time from VC files and you are Post Processing mode.

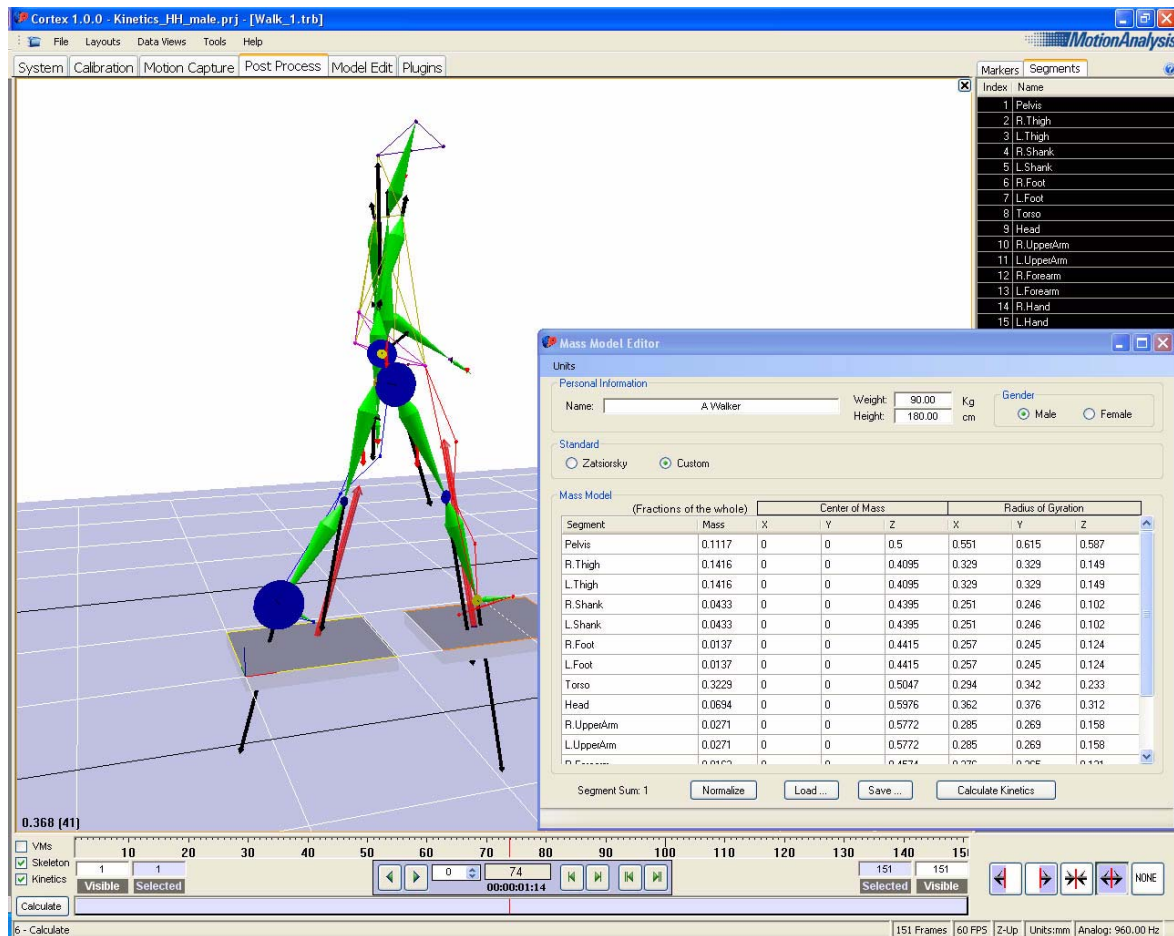
Figure 6-19. Forceplate Forces



	Newtons			mm			Newton-mm
	FX	FY	FZ	X	Y	Z	MZ
1	-46.95	3.37	372.32	-91.96	-8.02	84.00	-1785.22
2	70.15	-0.13	421.55	-615.77	-12.11	84.00	876.85

Mass Model Editor

Figure 6-20. Mass Model Editor



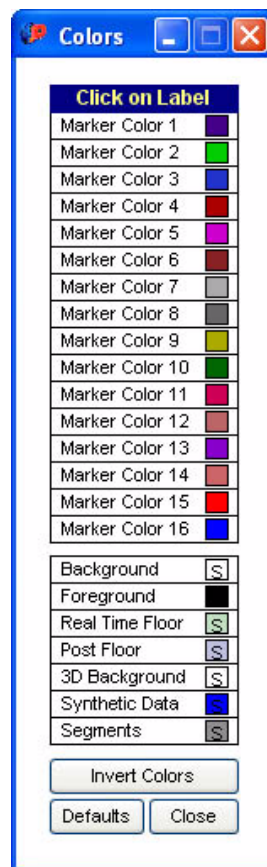
The Mass Model Editor shows, in tabular format, how the mass is distributed throughout the segments of the current skeleton model.

For more information, refer to the *Kinetics for Cortex User's Manual*.

Colors...

The Colors window allows you to choose RGB colors for the markers, segments, background, foreground, Real-Time floor, and post floor for your project. Refer to [Figure 6-21](#).

Figure 6-21. Colors Form



To change the color for any item, just click on the colored box and a color palette window opens up. This lets you choose the color blend for that marker. The colors are stored in the **Colors.xml** file that is part of the user's profile.

Misc

Replace Loaded Analog Channel Names—Occasionally, there are situations where an incorrect analog channel name can occur. Examples of this can be seen when analysis software (like OrthoTrak) requires specific muscle names for the analysis. In these cases the ability to go back and rename the problem analog channel is required.

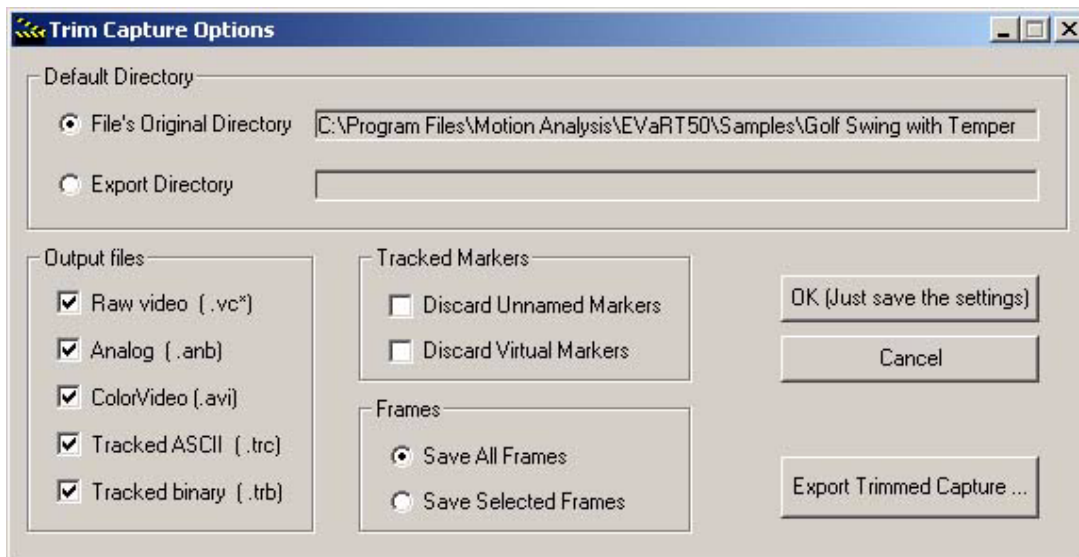
To Rename Analog Channels, follow these steps:

1. Load a Project file.
2. Load a Tracks (.trb/.trc) file.
3. Select **Data Views > Analog Display**.
4. In the existing project file, go to **System > Analog**, and change the name of the analog channel you wish to modify.
5. Save the project. You may want to save it as a different project name.
6. Go to the Tools menu and select **Misc > Replace Loaded Analog Channel Names**.
7. To see the replaced name, reload the project file you are working with. This will show the changed name in the Analog display sidebar.

Note: To save this change to the analog file, you must follow the next steps exactly.

8. Select **File > Trim Capture W/Options**.

Figure 6-22. Trim Capture W/Options Interface



9. Under the Trim Capture W/Options, choose **Save Selected Frames**. Make sure you have highlighted all the frames that you want by using the middle mouse button to highlight an area in the Post Processing window, or by using the Select All Frames button in the lower right corner.

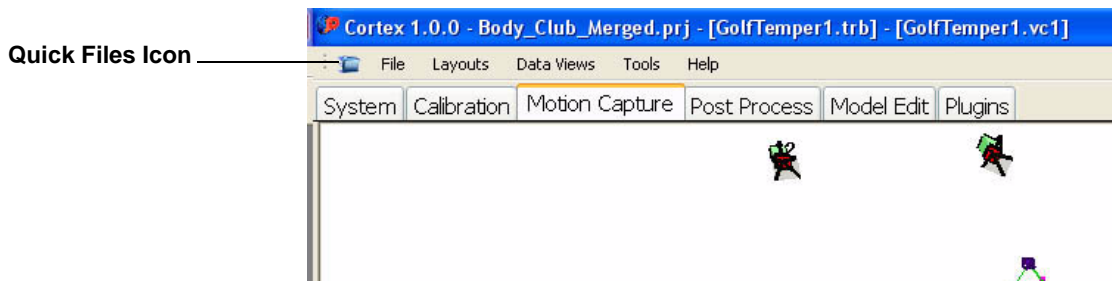
10. Press the **Export Trimmed Capture** button, and type in the filename you want or keep the current one.
11. If you have more files that need to have the analog channel names replaced, you will need to repeat steps 2 through 10.

Record—The record function under the Tools menu item starts the recording of a data capture. It is the same function as the Record button found on the **Motion Capture > Output** panel and the F12 function key.

Quick Files

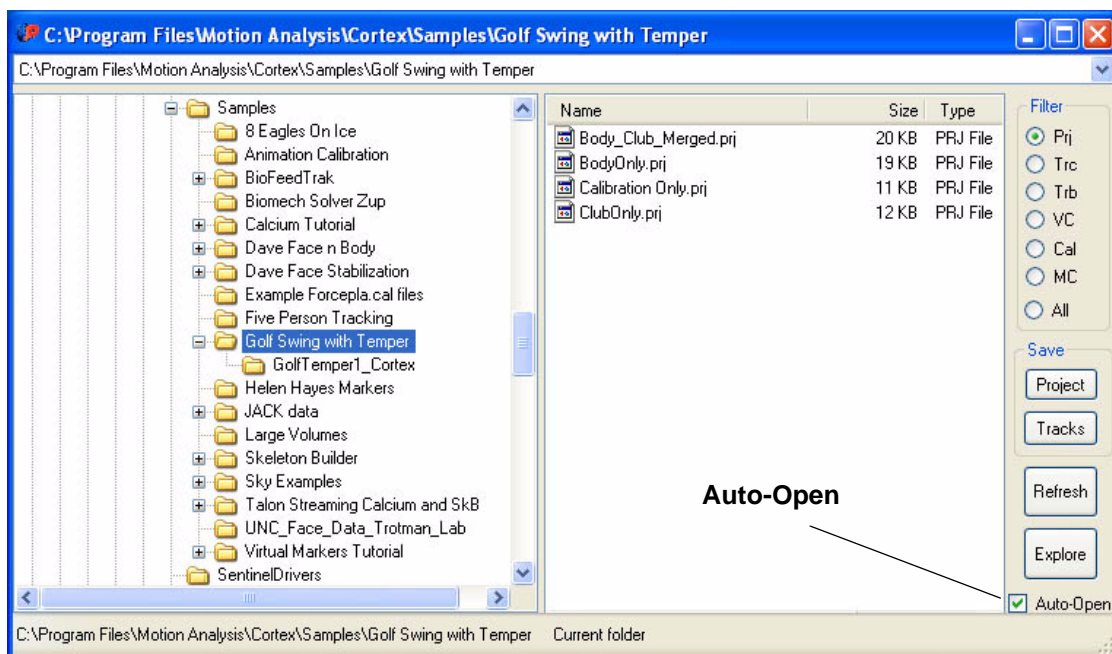
The Quick Files function allows you to easily navigate through frequently used directories, making it easier to address and manage motion capture sessions. Refer to [Figure 6-23](#).

Figure 6-23. Quick Files Icon



The Quick Files window will automatically open upon launch of the Cortex software if the Auto-Open check box is active.

Figure 6-24. Quick Files Interface



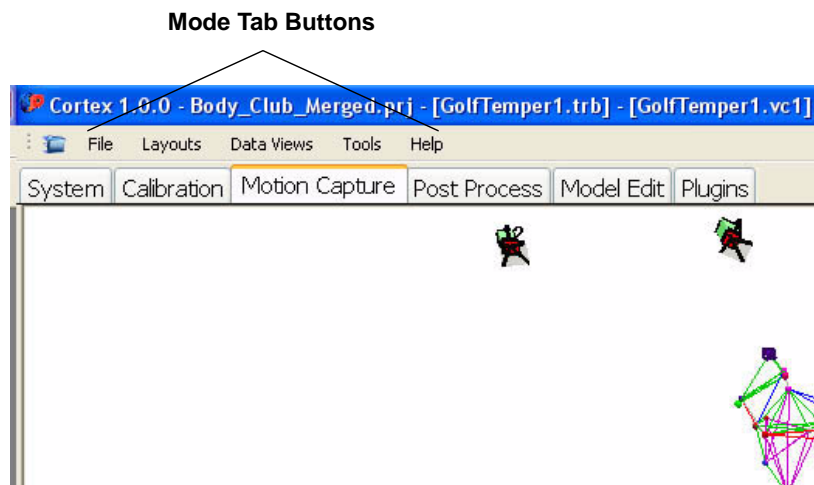
Help Menu

The Help menu provides information about the software, along with shortcuts, interface panel information, a searchable on-line manual, and access to the video tutorials.

Mode Tab Buttons

These buttons are arranged to guide you through a motion capture session in a phase-oriented order. Refer to [Figure 6-25](#). The first three mode buttons (System, Calibration, and Motion Capture) activate Real-Time mode and present you the necessary tools to successfully capture motion data.

Figure 6-25. Mode Tab Buttons



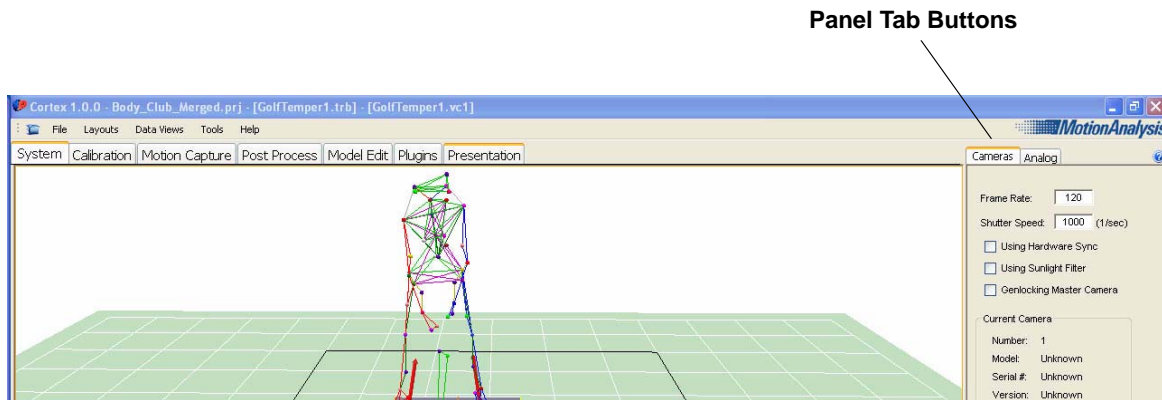
The fourth button, Post Process, activates Post Process mode and transforms **Cortex** into a tracked data editing tool.

The final two buttons, Model Edit and Plugins, are mode-less function buttons that present various tools without switching the program between the Real-Time mode and Post Process mode. Model Edit is used to define markers to create linkages.

Panel Tab Buttons

These buttons give you access to the various tools specific to the different phases of the motion capture session.

Figure 6-26. Panel Tab Buttons



Status Bar Messages

This feature, located in the lower left corner of the user interface, provides the status and confirmation of the software in its current processing state.

Real Time Dashboard

Refer to [“Real Time Dashboard” on page 6-40.](#)

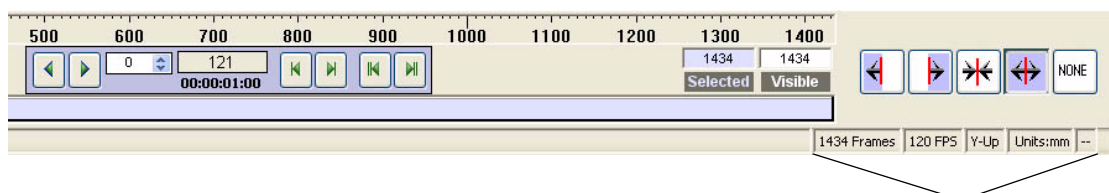
Information Center

The Information Center gives the following information for the current motion capture project (from left to right):

- Cell 1: Number of frames in the current data set
- Cell 2: Frame rate in frames per second
- Cell 3: Up axis (e.g. Y up, Z up)
- Cell 4: Calibration units (e.g. mm)
- Cell5: Analog sample rate (samples/sec)

If you leave the mouse pointer over the message, its definition will pop up.

Figure 6-27. Information Center



Information Center Cells 1 through 5

PRJ Files

PRJ files, or project files, are the main files used to set up and save all the variables involved with capturing data in **Cortex**. Every motion capture session must have a project file containing all system settings, equipment parameters, and other information related to the project. This file contains both equipment parameters common to many different setups and calibration values unique to one particular session. Among the data found in a project file are:

- the camera setup
- the marker set
- calibration setup and results
- linkages between markers
- SkB (Skeleton Builder) segment definitions, coordinate systems, and hierarchies (optional)—refer to [Chapter 12, Skeleton Types](#)
- MoCap Solver segment definitions, joint types, and hierarchies (optional)
- camera type and parameters
- tracking parameters

Note: Data trials are stored in separate trial files.

In most cases, you will begin a session by loading an existing project file, editing it as necessary, and saving it in the directory where the motion data is to be saved. Any time you calibrate the system or edit project parameters, you should save the project file to disk to retain the new information.

Cortex can read project files from older **EVaRT** software versions for backwards compatibility.

Note: Multiple PRJ files should be saved in the same file directory with care.

Important

Project files contain ASCII data and it may be useful to view them using any text editor, however, you should not edit them in a text editor as that can result in a corrupt file.

Viewing Sample Data

To become familiar with **Cortex**, we will start by replaying some sample data. This is done by loading a sample project found in the Samples directory.

Sample Data Set

1. From the Menu Bar, select **File > Load Project...**
2. Navigate to:
Program Files\Motion Analysis\Cortex\Samples\GolfSwings with Temper.
3. Double-click on **Body_Club_Merged.prj** to load the project.

Having loaded the sample project, we will now load the related data files.

1. Select **Raw Files** on the Real Time Dashboard.
2. Click on **GolfTemper1.vc1**.

Click the **Run** button on the Real Time Dashboard. At this point the action on the screen is a simulation of a live motion capture session. **Cortex** is processing the data from the stored raw video file, color video file, and analog file generated by the force plates. If this were an actual real-time capture session, the action on the screen would be similar, but the data would be coming directly from the cameras and force plates.

Note: Loading a VC file or TRB file will automatically load any saved ANB files captured during this session of this particular project. The ANB files are comprised of analog forceplate data in this example.

Having loaded all of the related data files, we can now exercise all of the six different Graphics Panes available to us. We will now look at four simultaneously.

1. From the Menu Bar, select **Layouts > 4 Panes**.
2. Left-click on the empty lower-left pane. This action will select this pane.
3. Press **F1** on the keyboard or choose **View > Color Video**.

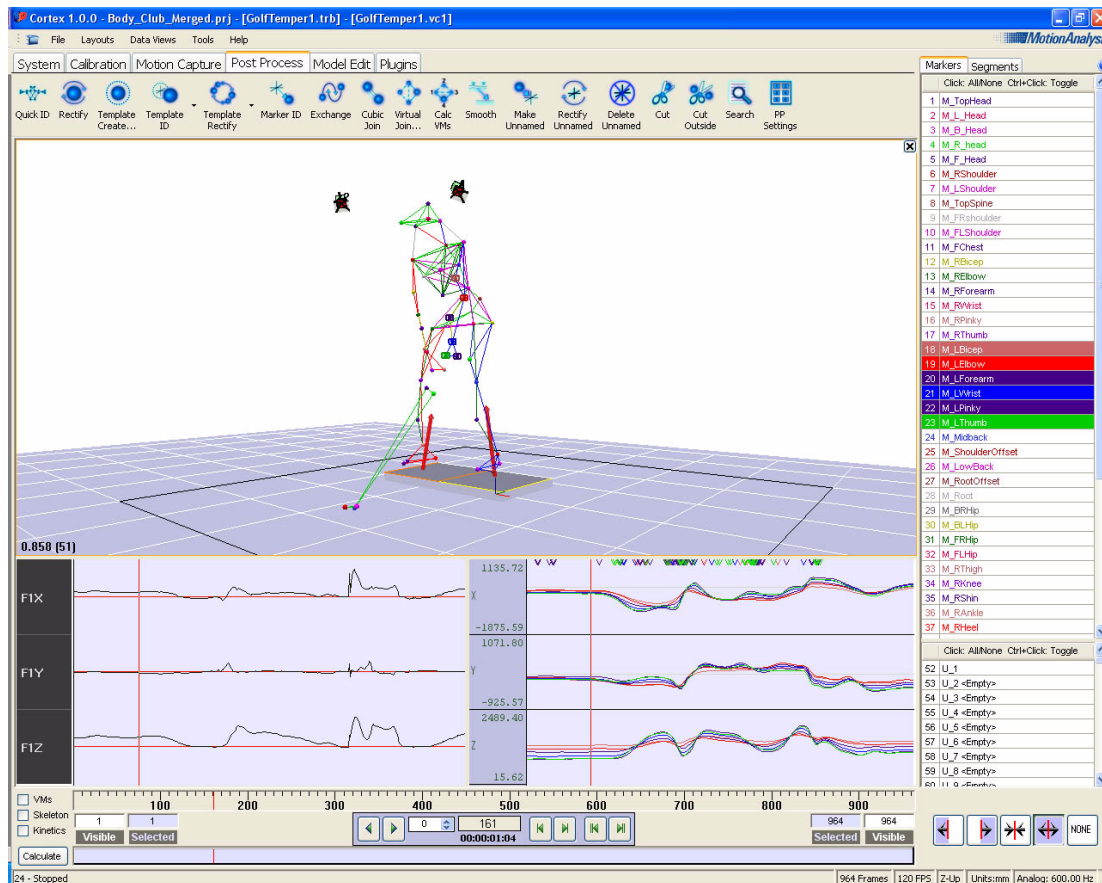
Note: If you are interested in the Color Video option, contact your Motion Analysis sales representative.

4. Left-click on the empty upper-right pane.
5. Press **F2** on the keyboard or choose **View > 2D Display**.
6. Left-click on the empty lower right pane.
7. Press **F5** on the keyboard or choose **View > Analog Display**.

The displays can be controlled by hand if you click **Pause**, click on the FIFO slider on the Real Time Dashboard, and then drag from side to side. The “First In First Out” FIFO slider can only manipulate the 256 frames of data that are currently stored in the FIFO memory space (but not all of the data in the data set).

Figure 6-28 should be similar to what you see on your screen.

Figure 6-28. Viewing the Pre-Recorded Data



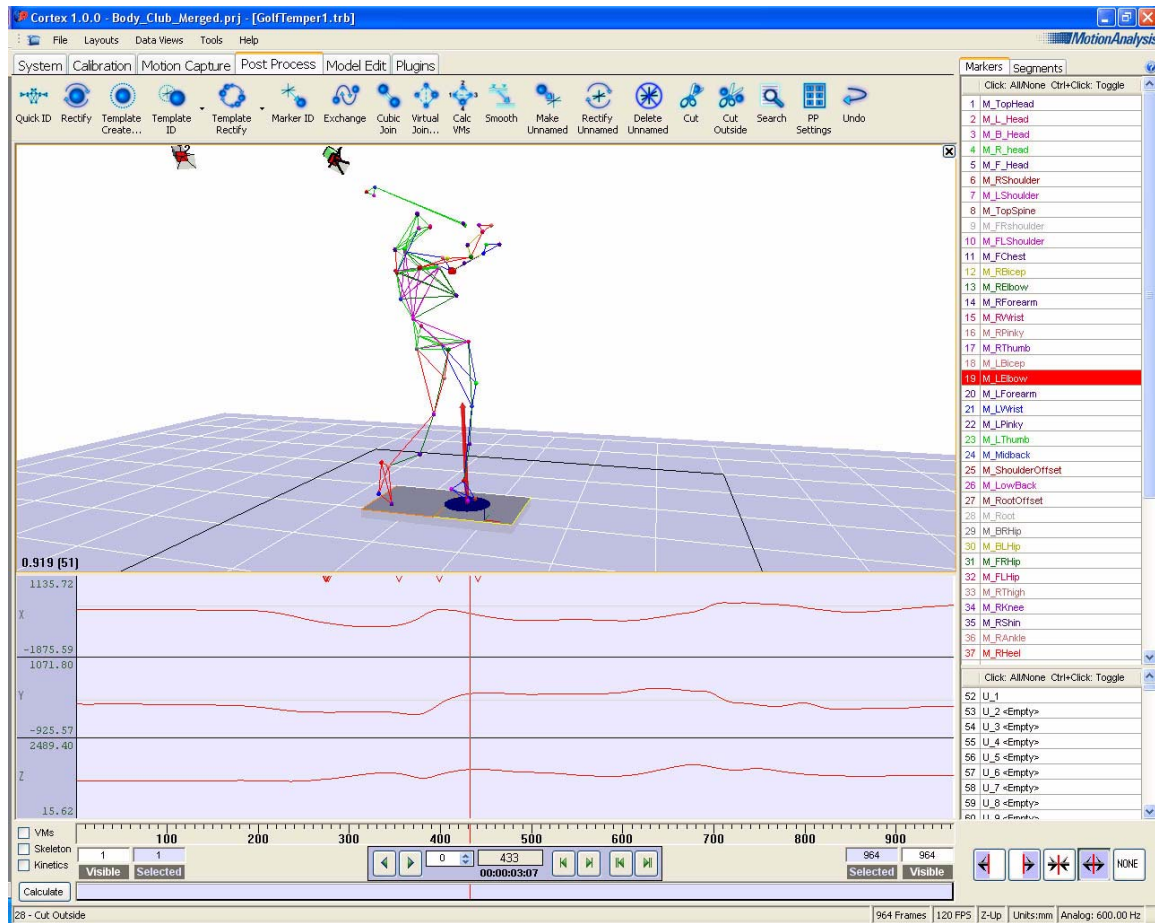
Cortex displays data somewhat differently when in Post Process mode. For instance, the analog displays become static graphs rather than having the oscilloscope style seen during collection and replays of raw data. The 2D Display and Skeleton Graphs become entirely unavailable but the XYZ Graphs become available. 3D stick figure images can be rendered for all of the data set rather than just the 256 frames available in Real-Time mode.

Post Process mode allows you to edit the tracked data generated during a motion capture session. Editing can be performed upon groups of markers or one marker at a time.

1. From the Menu Bar, select **File > Load Tracks File...**
2. Double-click on **GolfTemper1.trb**.
3. Leave the 3D figure currently in the top pane in the 3D view.
4. From the Menu Bar, select **Layouts > 2 Panes: Top/Bottom**.
5. Left-click on the bottom pane. This action will select this pane.
6. Press **F4** on the keyboard or choose **Data Views > Marker XYZ Graphs**.
7. Select marker 15 on the MarkerSet panel (right side).

The two Graphics Panes used, often simultaneously, during a Post Process editing session are the 3D View and the XYZ Graphs shown here. Notice that the Post Process Dashboard has replaced the Real Time Dashboard. The data shown in the XYZ Graphs represents the X, Y, and Z coordinates of the selected markers throughout the capture period.

Figure 6-29. Viewing Tracked Data in Post Process Mode



Pop-Up Menus

The Graphics Panes have viewing options and associated tools that can be accessed through pop-up menus. In all cases, the pop-up menus are activated with a click of the right mouse button while the pointer is in the display region.

3D View Pop-Up Menu

1. From the Menu Bar, select **Layouts > 1 Pane**.
2. From the Mode Buttons, left-click on the Post Process button in order to be in the Post Process mode.
3. If the 3D View is not visible, press **F3** on the keyboard or choose **Data Views > 3D-View** from the Menu Bar.
4. With the right mouse button, click on the 3D View.

The 3D View pop-up menu and descriptions of the tools are shown in [Figure 6-30](#). These options are recorded in your INI file and are reloaded when you launch **Cortex**.

Figure 6-30. Post Process 3D View With Pop-Up Menu Items

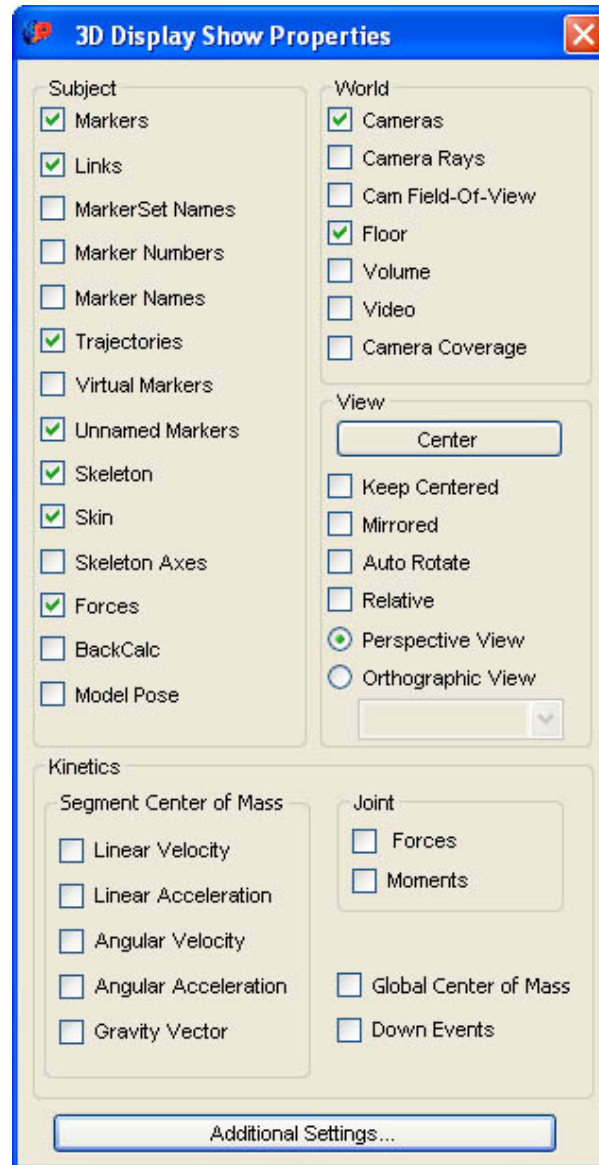
Show...	Show options—cascading menu
QuickID...	Quick ID the markers sequentially
Marker ID...	Marker ID the selected marker
Template ID	ID marker(s) based on current template
Rectify	Rectify marker(s) over the selected frame range
Hide Markers	Hide selected marker(s) from view
Unhide Markers	Show selected marker(s)
Make Unnamed	Make selected marker(s) unnamed
Create Template...	Create Template
Cut Inside	Cut data in selected frames from the selected marker(s)
Cut Outside	Cut data outside of the selected frames from the selected marker(s)
Exchange	Exchange data between two markers over the selected frames
Smooth	Smooth selected marker(s) over the selected frames
Join - Cubic	Join selected marker(s) over selected frames using cubic splines
Join - Linear	Join selected marker(s) over selected frames using linear interpolation
Join - Virtual	Create a temporary virtual marker to fill in missing marker data
Undo	Undo last action
Search	Search data set for spikes and/or gaps as defined in the Options panel

To see the 3D View options:

1. Choose the cascading **Show** item and another pop-up list will appear. Several of the view options will have check marks next to them indicating they are active. All of the Show items in the Show list are considered User Preferences and get stored to the **Cortex.ini** file when you exit the program.

2. Choose items from the Show options menu and see the effects. These options are saved in profile settings and are reloaded when you launch **Cortex**.

Figure 6-31. 3D View Right-Click Pop-Up View Options



Refer to the following page for descriptions of these options.

Subject

Markers—Displays all markers in the 3D View; Options... for length.

Links—Displays all links between markers in the 3D View.

MarkerSet Names—Displays the prj file name over the marker cloud.

Marker Numbers—Displays the ID numbers for all markers.

Marker Names—Displays the marker names (over the marker).

Trajectories—Displays the marker trajectories (in PP mode only).

Virtual Markers—Displays the virtual markers.

Unnamed Markers—Displays unidentified markers.

Skeleton—Displays the skeleton segments when they are defined.

Skin—Displays a selected skin.

Skeleton Axes—Displays the RGB/XYZ orientation for each bone.

Forces—Displays the forceplate and force vectors off the forceplates.

BackCalc—Displays the marker locations as defined in the T-pose for Calcium skeletons.

Model Pose—Displays the model pose as defined in Create Template.

World

Cameras—Displays the motion capture cameras in the 3D View.

Camera Rays—Displays which cameras can see the selected marker(s).

Cam Field-Of-View—Displays the camera view for the selected camera(s) based on the focal length as set in the Lenses/Orientation tab.

Floor—Displays the virtual floor in the 3D View.

Volume—Displays the capture volume.

Video—Displays the digital video layered with the 3D View.

Camera Coverage—Displays the coverage of all cameras within the capture volume.

View

Center—Centers the display on the selected marker.

Keep Centered—3D View center follows the selected marker.

Mirrored—Mirrors the marker in the field of view.

Auto Rotate—Rotates around the capture area by selecting Play.

Relative—Toggles display to view from selected markers.

Perspective View—Selects the normal display options (adjust with mouse).

Orthographic View—Displays a flat view from either the X, Y, or Z axis perspective.

2D Camera View Pop-Up Menu

The 2D Display renders raw camera data as blobs and/or marker centroids. You can choose to see the centroids either with or without lens correction. To see the marker data as viewed from any one of the cameras or multiple cameras simultaneously,

1. Press **F2** on the keyboard or choose **View > 2D Display** from the Menu Bar.
2. Choose one or more cameras with **Ctrl** + click or **Shift** + click on the green camera buttons on the Real Time Dashboard or press **All On**.

To see the 2D Display options, right-click on the camera view 2D Display.

Figure 6-32. 2D Camera View Pop-Up View Options

Delete Mask	Deletes selected mask
Delete All Masks	Deletes all masks in the 2D display for the selected camera
Auto Mask	Draws a mask around all items in the field of view
Software Masks Only	Allows the capture of raw data with any masks in 2D display
Show Raw Data	Toggles black raw data blobs
Show Raw Centroids	Toggles red, raw centroid crosses
Show Corrected Centroids	Toggles centroids corrected from lens distortion
Show Names	Toggles marker names
Show Numbers	Toggles marker numbers
Show Volume Floor	Toggles outline of volume floor
Smear Display	Leaves smeared paths of the markers
Show Greyscale Image	Displays the greyscale image for the selected Raptor camera
Reset Zoom	Resets the selected camera to standard zoom settings
Reset Greyscale Region	Resets the Raptor greyscale image to full camera view
Reset Zoom For All	Resets all cameras to standard zoom settings

Show Greyscale Image

This shows an 8-bit greyscale image from a 10-bit greyscale sensor. Markers (blobs) appear in blue for better separation. This used only with Raptor class cameras.

2D Display Shortcuts

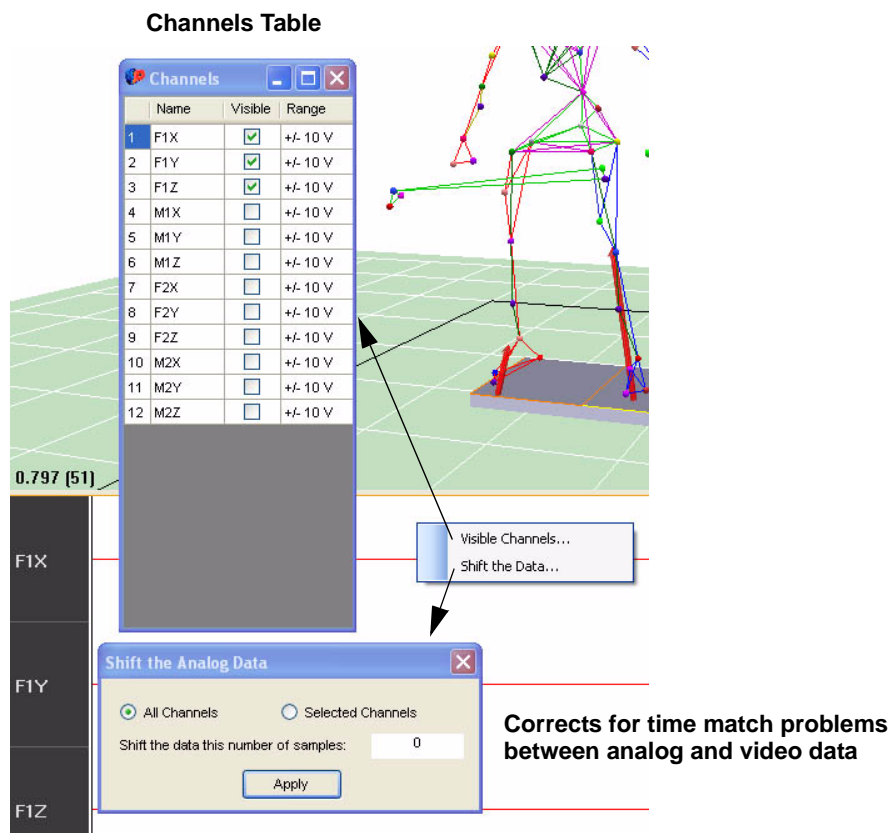
- Double-clicking on a specific camera will show only that camera's view. Double-clicking again will return to all camera views.
- Selecting **Shift + double-click** selects the view for only the enabled cameras within the system.

Analog Display Pop-Up Menu

For users collecting analog data from force plates, an Analog Display provides graphs of output from up to 192 analog channels. You can view any combination of channels at the same time. As a convenience, the Analog Display allows you to resize the label panel on the left side of the screen to accommodate long channel names. To open the Analog Display and modify the number of visible channels:

1. Press **F5** on the keyboard or choose **Data Views > Analog Graphs** from the Menu Bar.
2. With the right mouse, click on the Analog Graphs.
3. From the pop-up menu, choose **Visible Channels...**
4. Left-click on any one of the check marks in the Visible column.
5. Press **Shift** + click in the **Visible** column to toggle multiple channels.
6. Click directly on the **Visible** header cell to toggle all of the channels at once.

Figure 6-33. Analog Display With Right-Click Pop-Up View Options and Channels Table



XYZ Graphs Pop-Up Menu

Post Process mode allows you to edit tracked data. The XYZ Graphs displays the positions of each marker in each frame. It also lets you select and edit those markers in any frame. A complete discussion of editing tracked data can be found in [Chapter 10, Post Processing Panel](#). To see the XYZ Graphs and the pop-up menu of tools and view options:

1. Press **F4** on the keyboard or choose **View > XYZ Graphs**.
2. When in Post Process mode, right-click on the XYZ Graphs.

Figure 6-34. Post Process XYZ Graphs With Pop-Up View Options and Tools

Zoom Frames In	Zoom into the current frame range
Zoom Frames Out	Zoom out from the current frame range
UnZoom	Reset the amplitude display
Auto Scale	Auto scale to visible channels
Uniform Scale	Applies uniform scale to all three X, Y, and Z panels
Select All (Named) Data	Select all frames in the data set
Show Residuals+Camera	Show Residuals and Cameras plots
Quick ID...	Quick ID the markers sequentially
MarkerID...	Marker ID the selected marker
Template ID	ID marker(s) based on current template
Rectify	Rectify marker(s) over the selected frame range
Hide Markers	Hide selected marker(s) from view
Unhide Markers	Show selected marker(s)
Make Unnamed	Make selected marker(s) unnamed
Create Template....	Create Template
Cut	Cut data in selected frames from the selected marker(s)
Cut Outside	Cut data outside of the selected frames from the selected marker(s)
Exchange	Exchange data between two markers over the selected frames
Smooth	Smooth selected marker(s) over the selected frames
Join - Cubic	Join selected marker(s) over selected frames using cubic splines
Join - Linear	Join selected marker(s) over selected frames using linear interpolation
Join - Virtual	Create a temporary virtual marker to fill in missing marker data
Undo	Undo last edit on marker data
Search	Search for spikes and/or gaps; as defined in the Search tab

Digital/Reference Video Option (EVaDV Software)

The color Digital Video option allows you to record a time-matched Reference Video along with your motion capture trial on a separate computer. With this option, you will record a time-matched color video AVI file with the same trial name in your motion capture folder. A separate computer is used in order to not burden your **Cortex** Host computer, which is an issue if your computer is too slow for the number of markers being tracked. For single person captures, you may connect the DV Camera directly to the **Cortex** Host computer. In this case, the **EVaDV** software is not needed. It is built into the **Cortex** software. You can run **EVaDV** on one or more computers and then capture multiple AVI files (multiple views). They will all have the same AVI file name. You may experience a small delay in frames from the **Cortex** software and the **EVaDV** software when capturing. The Color Video display has a pop-up menu with one item, Adjust Frame Offset. This allows for time-matching data streams.

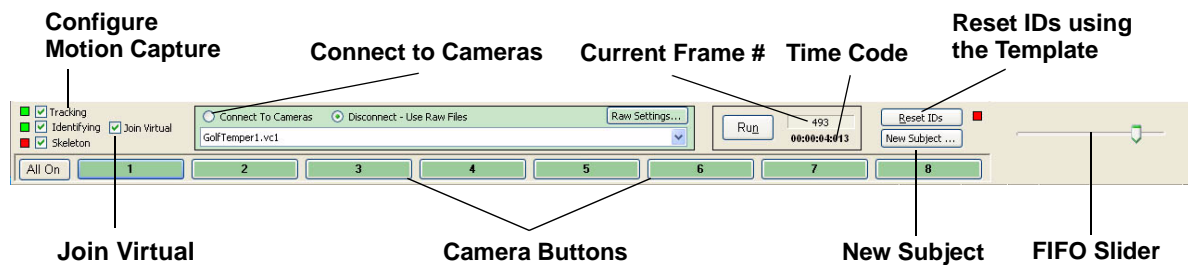
Note: The **EVaDV** software option is not to be confused with the AVI function in the **Motion Capture > Output** panel. This function creates an AVI file when you collect a trial within **Cortex**.

Real Time Dashboard

The Real Time Dashboard is available when **Cortex** is in Real Time mode as opposed to Post Process mode. When you are capturing data in real-time, this dashboard provides the controls to manage a motion capture session. It also supports the replay and tracking of previously recorded data with a simulation of real-time from the raw VC camera files.

Note: To help you distinguish between the two modes (Real Time and Post Process), the dashboard and the floor color changes.

Figure 6-35. Real Time Dashboard



Cortex Dashboard Camera Button Colors

The Real Time Dashboard camera buttons inform you of the following:

Green	Camera is completely calibrated
Light green	Camera is completely calibrated and is selected
Yellow	Camera has undergone seed calibration but not wand calibration
Light yellow	Camera has undergone seed calibration and is selected
White	Camera is not calibrated
Dark grey	Camera is inactive—A right mouse click on the camera number will enable and disable that camera.

Note—Camera #1 must remain enabled when you collect data to make sure you have a selectable VC1 file.

Tracking

The Tracking check box triangulates (tracks) the markers from frame to frame. You might want to uncheck this if your computer is not fast enough to calculate the marker coordinates or if your system is not calibrated for any reason. You can still collect raw VC files and track them at a later time. Collecting raw VC files is the highest priority thread in the motion capture Record mode to ensure that you do not lose your raw data.

Identifying

The Identifying check box identifies and names the tracked markers according to the current template. If you do not have a template, it is best to disable this function to keep the software from attempting to ID the data.

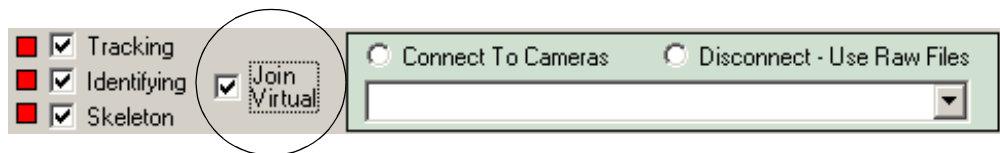
Skeleton

The Skeleton check box is set to calculate the skeleton using the currently active skeleton model.

Join Virtual

Join Virtual is an extremely powerful editing tool used to fill gaps in marker data with simulated data based on the relationship (positional interpolation) with other markers on or near the particular problem segment. This positional interpolation is defined by the settings in the **Tools > Virtual Marker Definitions** function.

Figure 6-36. Join Virtual Check Box

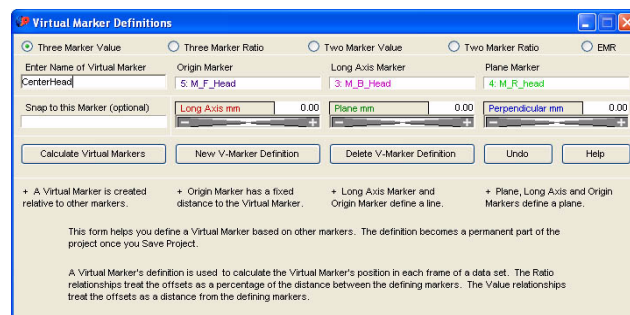


The concept behind the Join Virtual and the Virtual Marker definitions are the same and are much more stable and more useful than the classic Rigid Body data filling mechanisms. The reason is that you get to choose two sets of three markers, in decreasing importance, that determine the replacement data. These three markers are:

1. the Origin Marker
2. the Long Axis (Y) Marker
3. the Plane (XY) Marker

The two sets of virtual marker definitions allow you to continue generating virtual marker data if one of the definition markers is not being tracked. For the Join Virtual function to work properly, you will need a minimum of three different support markers among the six spots to fill. If you are in Streaming mode from cameras or VC files, the first definition set is used. If you are in Post Process mode, you may choose which definition set works best.

Figure 6-37. Virtual Marker Definitions



When running live, the Join Virtual tool only uses the first VM Join definition of the two that you are allowed. However, 4 passes are made over the list on each frame so that if a definition depends on another then after the first pass the second marker is reconstructed so that the first marker can be reconstructed on the second pass. It also works this way in Post

Process mode when you have multiple markers selected and do a Join Virtual function.

Streaming vs. Post-Processing

The Real Time Join Virtual check box (in the Real Time dashboard) eliminates what might be seen as a possible “pop” on the frame when the real marker re-appears. At that time, the Virtual Marker filling the gap is no longer used.

In the post-processing Join Virtual mechanism, the offsets between the marker to join and the Join Virtual Origin Marker are measured both at the start of the gap and the end of the gap, and a linear interpolation is used for all in between data points. The result is always a perfectly fluid transition on both ends of the gap.

The Join Virtual mechanism is a powerful tool in creating and editing data quickly with good results. It is the result of working with our customers to define and develop techniques to get good motion capture data quickly and efficiently.

Reset IDs Button

The Reset IDs button forces the current template to be used for that specific frame. It is used when a marker is misidentified. Press this if you see markers that are incorrectly identified. If this fails to fix the problem, you may need to create a new template or adjust the marker set to be less symmetric. After pressing the **New Subject...** button and with the Pose ID window open, the Reset ID button will use the current marker pose to identify the markers.

New Subject Button

This feature allows the software to automatically identify markers based on a Model Pose that you create when you make a template. The result is that if you use the same marker set repeatedly, you will not have to ID the new person each time the marker set is used. The marker identification is automatic and instant, saving you time.

The Model Pose has its own kind of generalized template that is used to automatically identify a new person when they appear in the field of view. It saves the steps of using the Quick ID feature to identify a new person in order to make a template for them. The Template ID feature works in the Real-Time mode when you are connected to cameras or it works when you are tracking the data from VC files after the collection.

Note: The current template is size specific, so a new person or a new arrangement in the markers will not generally work for automatically identifying the markers.

When you click **Update Template**, it also updates the Model Pose.

To use the **New Subject** button, use the following procedure:

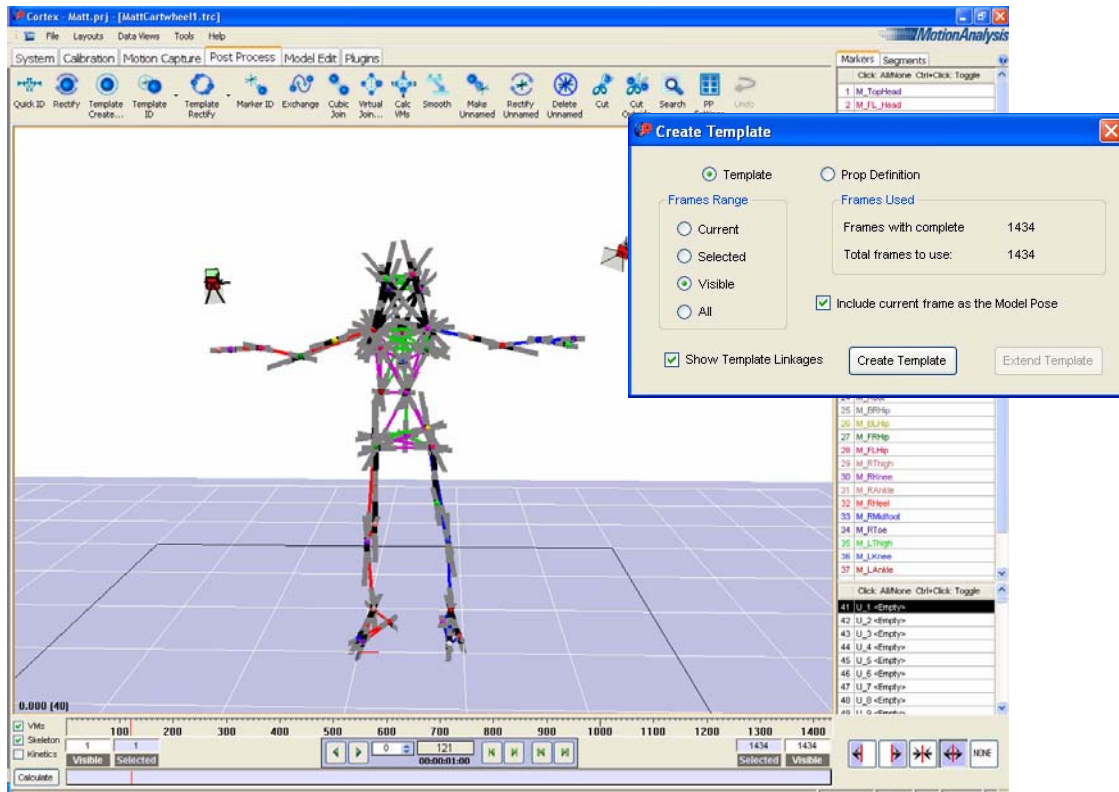
1. Get a Range-of-Motion Trial.
 - a. Get a good range of motion trial for your current tracks in Post Process, Quick ID, and edit so there are no mistakes or marker switches. The data does not have to be highly complex, but it should represent the minimum and maximum stretching for all

limbs. Jumping Jacks are a good example of the kind of dynamic motion that has worked well and does not obscure the markers or require editing. For simple walking motion, a single walking trial will suffice.

- b. Select one frame that represents a standard or neutral pose position. This can be with the arms down or the arms out, feet apart or together, but where no markers will be hidden. You will want it to be a standard position that the next person will be able to repeat quickly and simply. Have them face a certain direction that will also be easily repeatable for the next person. (along the +X axis for example).
2. Create a Template.
 - a. Select **Post Process > Create Template**.
 - b. Select Body Template and check the box **Include current frame as the Model Pose** and select the correct range of frames where you have good data.
 - c. Save your project file which now has a new feature called the Model Pose stored in it. You may want to use the word "Pose" in the project file name to distinguish it from earlier versions without the pose, but that is not necessary. The normal template will also work for this person.

Note: You can choose to see the Model Pose in your current project by right-clicking in the 3D View and then selecting **Show > Show Model Pose**.

Figure 6-38. Create Template Window Showing Template Linkages



Getting Auto ID to Work: Tuning and Updating the Template

If the person who was used to create the template moves some markers, or if a new person comes out with the same marker configuration (but in slightly different locations), you will want to update the template to the new marker locations. This will make sure that your RealTime tracking and the Template ID and Template Rectify functions in Post Process will be at an optimum performance level.

Activate the Motion Capture tab and then select **Connect to Cameras** (or select the VC files) and then select **New Subject....** This will bring up the Pose ID dialog box (below) and the Model Pose stick figure appears in the 3D View.

Figure 6-39. New Subject... Interface

At this point the old Template ID feature is not working, but instead the generalized automatic Pose ID feature is looking to identify the unnamed markers. As soon as the ID is recognized and catches, you will see two stick figures; the static one from the Model Pose and one that is the newly ID-ed person that is moving. If the person is not ID-ed right away, have them face the same direction and assume the same general posture as seen in the Model Pose. The Auto ID feature works as long as the person is facing within about 45° of where the Model Pose was recorded.

The status display in the lower left tells you how fast the ID process took. A small number is a fast ID, a bigger number would be slower, but still working. Using the Reset IDs button on the lower right will force the software back to the Pose ID if something gets switched and you want to correct it.

Figure 6-40. Pose ID in Message Center and Reset IDs Button



Updating the Template also updates the Model Pose, so before clicking the **Update Template** button, you should again get your new person into something close to the Pose position. The changes can be saved in your project file if you want.

After you Update the Template, the template is then re-sized to the new person's limb lengths and marker placements. Note that the changes in the lengths, as recorded in the range of motion trial is still saved, so that you will not need to do another range of motion. The new template should work well for many sizes and marker adjustments using the same marker set.

Recommended Procedure

1. Create your own library project file for the marker set. This library contains your markers set with the template created from your range of motion TRB files, and your Pose ID.
2. When creating the library project file, start with the range of motion TRB type file in the neutral position, facing +X, arms down, feet slightly apart to show all the markers. That would be frame 1 to make the Pose ID easy to find. When you create the Template for your library file, you would have Frame 1 selected as the current frame and you would check **Include Current Frame as Model Pose** for frame number 1. You should only need to do this once per marker set.
3. When the New Subject comes into the volume, they should be standing in the Pose position facing the same +X¹ and you select **New Subject...** The PoseID will show and you should then select **Pause**. The stick figure should snap to, looking like the Model Pose.
4. Update the template, then select **Pause**. Check that the ID is correct and that the position is similar to the Pose. You can use the FIFO slider to make adjustments. Save the project file and then select **Run**.

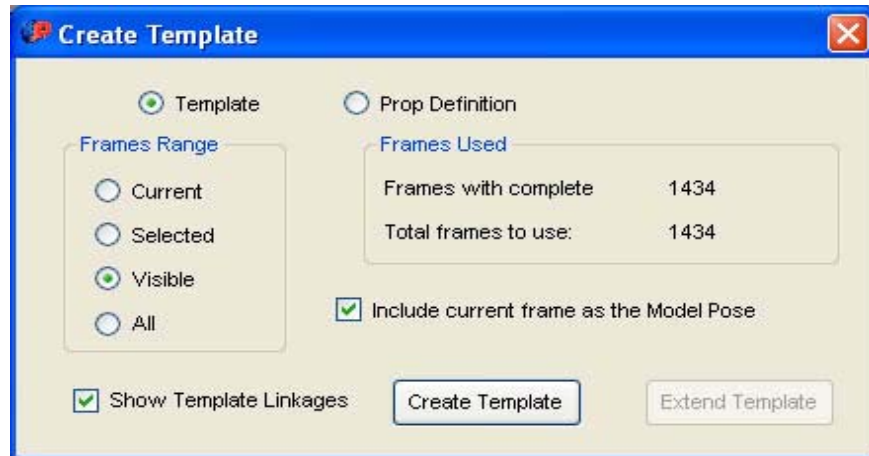
Get a Range of Motion Trial and Make a Template

1. You will first want to obtain a good range of motion trial and set it as your current tracks in Post Process.
2. You will then need to Quick ID and edit the trial, if needed, so there are no mistakes or marker switches. It does not have to be overly complex, but it should represent the minimum and maximum stretching for all limbs. Jumping jacks is a good example of the kind of dynamic motion that has worked well and does not obscure the markers or require editing. For simple walking motion, a single walking trial will be sufficient.

¹Facing +Z or any other direction also works as long as Pose ID is used in the same orientation

3. Select one frame that represents a somewhat standard or neutral pose position. This can be with the arms down or the arms out, feet apart or together, but where no markers will be hidden. You want it to be a "standard" position that the next person will be able to repeat quickly and simply. Have the subject face a certain direction that will also be easily repeatable for the next person. (along the +X axis for example).

Figure 6-41. Create Template Interface



Create a Template

1. Select **Post Process > Create Template**.
2. Select **Body Template**.
3. Activate the **Include current frame as the Model Pose** check-box and select the correct range of frames where you have good data.
4. Save your project file which now has a new feature called Pose stored in it. You may want to use the word Pose in the project file name to distinguish it from earlier versions without the pose, but that is not necessary. The normal template will work for this person.

Using the Pose ID Feature

This feature allows the software to automatically identify markers based on a Model Pose that you create when you make a template. The result is that if you use the same marker set repeatedly, you will not have to ID the new person each time the marker set is used. The marker identification is automatic and instant, saving you time.

The Model Pose has its own kind of generalized template that is used to automatically identify a new person when they appear in the field of view. This generalized template depends on the person facing the same direction as the stored Pose ID and having the markers in the same general locations with respect to each other. You save the Pose ID from one frame of data and is saved in your project file. It saves you the steps of using the "Quick ID" feature to identify a new person in order to make a template for them. The Auto ID feature works in the RealTime mode when you are connected to cameras or it works when you are tracking the data from VC files after the collection.

The following are basic steps on how to use the Pose ID function.

Connect Cameras Button

The Connect Cameras button will activate all the cameras used in a motion capture session. A pop-up message announces the number and type of cameras, A-D units, color video cameras, and time code reader if present.

Disconnect - Use Raw Files Button

The Raw Video button will tell the program to simulate a live motion capture session from previously captured Video Camera (VC) files. It also allows you to tack and record to TRB or TRC trials for which you have raw VC files.

Raw Settings... Button

This button opens the **Settings > Playback** interface. For more information, please reference ["Playback Tab" on page 6-13](#).

Run Button

The **Run** button will start the streaming of live camera data or start the simulation of a motion capture session from existing raw VC files.

The **Run** button has the following functions:

1. If you are connected to the cameras, it starts the data steaming from the cameras. You are able to record the Raw Video VC files as set in the motion capture Output panel. Check your 2D views to be sure the cameras, masks, and thresholds are all set properly.

2. If **Enable Tracking** is checked, you will see the marker data appear in the 3D View. This requires that the system has been calibrated. You are then able to record VC files, and TRC or TRB files.
3. If **Enable Identifying** is checked, you will see the colored markers and the stick figures in the 3D View. This requires that the system is calibrated and a template is defined and operating. You are then able to record VC files, and TRB or TRC files.
4. Tracking from Raw Video files—If you are not connected to the cameras, you have the full range of option 2 or option 3 above from your previously collected Raw Video VC file.

Time Code Counter

The Time Code Counter displays the frame number in HH:MM:SS:Frame (hour, minute, second, frame) format. If you have the optional Time Code Reader card installed in your computer, this displays the current Time Code value when you are in the Motion Capture mode and connected to the cameras.

Frame Counter

The Frame Counter displays a count of the total number of frames in the data set.

Camera Buttons

You can select each camera by clicking on it's respective numbered button that is listed across the Real Time Dashboard. Clicking on a camera button will either activate or de-activate that camera for setup features.

Right-Click Camera Buttons

Right-clicking on any of the camera buttons will open a function menu with various commands for that specific camera. The menu and a description of each command is shown in [Figure 6-42](#).

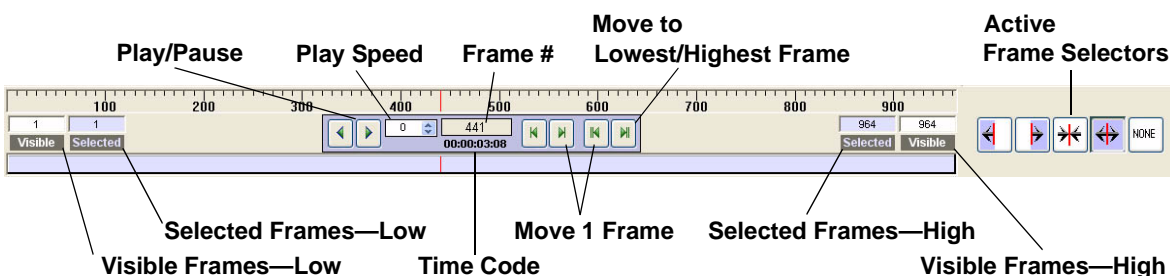
Figure 6-42. Camera Buttons Right-Click Menu

Enable	Enables the selected camera to capture data (if disabled)
Disable	Disables the selected camera from capturing data
Enable for Live Tracking	Enables the camera to capture and display data in Real Time
Disable for Live Tracking	Disables the camera from data collection in Real Time only
Sort by Position	Camera numbers are sorted in a counter-clockwise order
Sort by IPAddress	Camera numbers are sorted by IP address starting from lowest
Delete	Deletes the selected camera from the project file

Post Process Dashboard

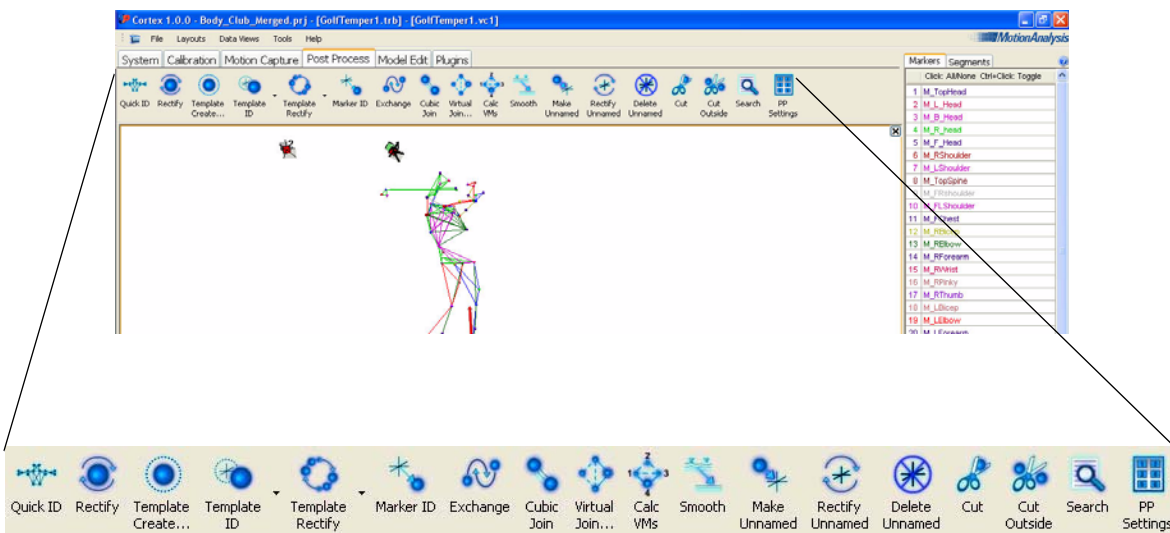
The Post Process Dashboard is available when **Cortex** is in Post Process mode as opposed to Tracking mode. After you have generated and saved tracked data, this becomes available to help manage a data editing session. It controls the range of visible frames and the range of selected frames to be edited. It also provides several controls for playing through the tracked data and choosing a current frame. This dashboard is described further in [Chapter 10, Post Processing Panel](#).

Figure 6-43. Post Process Dashboard



Post Process Tool Strip

Figure 6-44. Post Process Tool Strip



Post Process Tool Strip Icons

For a listing and description of all Post Process Tool Strip Icons, reference [“Post Process Tool Strip” on page 10-12](#)

Zooming, Rotating, and Translating in the 3D View

Zooming and translating a display can occur in both the 3D View and the XYZ Graphs. Rotating only occurs in the 3D View. Choosing

Help > Hot Keys and Tips from the Menu Bar will bring up an online table describing how these features work.

Zoom—In the 3D View, zooming is accomplished if you:

1. Hold the **Alt** key down.
2. Hold both the **left** mouse and **middle** mouse buttons down.
3. Move the mouse forward or left to zoom out and backward or right to zoom in.

Rotate—In the 3D View, rotating is accomplished if you:

1. Hold the **Alt** key down.
2. Hold the **left** mouse button down.
3. Move the mouse in any direction.

Translate—In the 3D View, translating is accomplished if you:

1. Hold the **Alt** key down.
2. Hold the **middle** mouse button down.
3. Move the mouse in any direction.

In the XYZ Graphs, time zooming is done in terms of frames (time) or amplitude. If you want to zoom in frames, there are two methods. 1) If one or no frames are selected, zooming is done relative to the current frame. 2) If two or more frames are selected, zooming is done relative to the selected frames.

Time Zoom Method 1

1. Click on the **Post Process** button among the Mode Buttons.
2. Press **F4** or choose **View > XYZ Graphs** from the Menu Bar.
3. Left-click on **None** in the lower right corner below the marker list. Now, no frames are selected.
4. Left-click anywhere on the XYZ Graphs to set the Current Frame which is indicated by the red line.
5. To zoom in, press the “Zoom Frames In” hot key (default is **I**) or by right-clicking in the XYZ Graphs and selecting “Zoom Frames In” from the pop-up menu.
6. To zoom out, press the “Zoom Frames Out” hot key (default is **O**) or by right-clicking in the XYZ Graphs window and selecting “Zoom Frames Out” from the pop-up menu.
7. Unzoom time: Double-click on the Time Slider on the Post Process Dashboard to zoom out completely making all frames visible. The Post Process Dashboard Visible boxes will now have a **1** and the highest frame number displayed.

Time Zoom Method 2

The second method of zooming in frames is described as follows:

1. Hold the **middle** mouse button down in the XYZ or the Analog Graphs.
2. Drag the mouse to the right or left to select any number of frames.
3. To zoom in, press the “Zoom Frames In” Hot Key (default is I) or by right-clicking in the XYZ Graphs window and selecting “Zoom Frames In” from the pop-up menu.

Amplitude Zoom

Zooming amplitude is done relative to the closest data point and frame nearest to the location you initially click on. You can optionally zoom into the data in the Current Frame regardless of where your mouse cursor is on the screen. This option is a User Preference and it can be set by launching the Options Form from the Post Process tab.

1. Hold the **Alt** key down.
2. Hold both the **left** mouse and **middle** mouse buttons down.
3. Move the mouse forward and backward.

In the XYZ Graphs, translating is accomplished if you:

1. Hold the **Alt** key down.
2. Hold the **middle** mouse button down.
3. Move the mouse in any direction.

Selecting Markers, Virtual Markers, Linkages, and Segments

Markers can be selected by the following:

1. Clicking on the markers seen on the 3D View
2. Double-clicking on the markers seen in the XYZ Graphs
3. Clicking on the markers listed on the Marker Grids (which are a part of the Post Process tab, the **Model Edit > Markers** panel, and the **Model Edit > Tree View**)

Note: All of the conventional **Shift** + click and **Ctrl** + click techniques to select multiple items are supported in this software.

Time Code

SMPTE Time Code and Cortex Overview

SMPTE Code reads as HH:MM:SS:FF, which is Hours:Minutes:Seconds:Frame. Frame numbers are 0 to 29 in NTSC and 0 to 24 in PAL. When you capture at a higher motion capture rate such as 60 or 120 Hz, there are multiple motion capture VC frames per color video frame. The software takes care of that so if you record your VC files at 60 Hz, the VC frame advances twice for every single frame advance in the color video when you play it back or step through the data. The SMPTE time code is visible on the Real Time Dashboard.

Using the Time Code Reader Option with Eagle and Hawk Cameras

Eagle and Hawk digital cameras can use the Time Code Reader (PCI version) card, installed into the **Cortex** Host computer. It reads the LTC (Longitudinal Time Code) from the RCA audio connector on the Time Code card, creating a **trialN.tc** file (time code) when you collect a **trialN.vcX** dataset. It is automatic if you have the Time Code Reader option (card) installed in your **Cortex** computer. There is a BNC type connector on the card as well; it appears that the Time Code Reader will genlock to the black burst video signal, but that is not needed. **Cortex** reads the current time code when the data collection is started and time stamps it into the TC file. The current time code also displays on the Post-Process Dashboard.

A simple test program called **TimeCodeReader.exe** is distributed with the latest **Cortex** releases for Eagle and Hawk camera users. It is a stand-alone program which launches, and in a small window reads the current value of the Time Code Reader in the **Cortex** Host computer. It is useful for testing to see if the Time Code reader is working. Without a card installed, it just leaves a blank display. With a time code reader card installed, it displays the current time code, static or not. When the time code starts to advance, you can immediately see it.

Using the Time Code in Post Processing

If you load a TRB or TRC file that has an associated TC file, then the Post-Processing Dashboard will lock the time code onto the time code display. You can step forward or backwards in time or push the play button and the time code reads accurately. If you switch back to the Motion Capture tab and are connected to the cameras, you will see the current time code. If you are in the Motion Capture tab and are not connected to the cameras, but using Raw Video Files, you will see the Time Code associated with the Raw Video File.

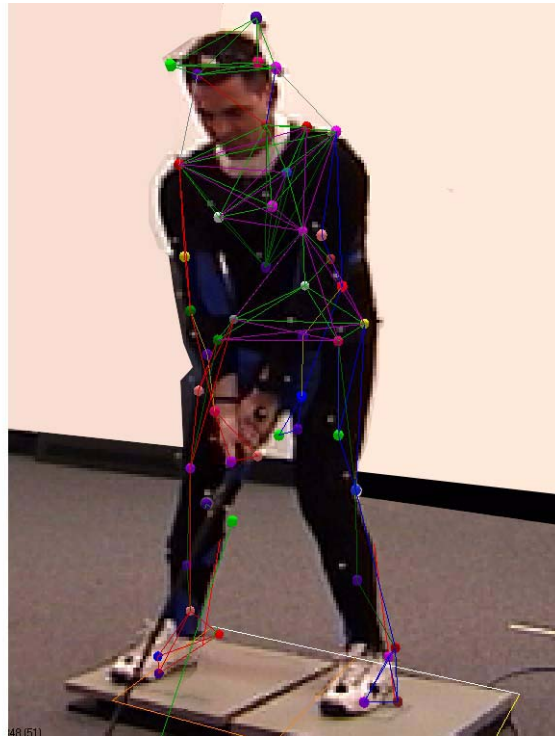
Time Code and the Digital Video (EVaDV) Option

The Digital Video option can be used with the Time Code. In our current software, the only way to record the time code is with the Time Code Reader card. Some cameras have Time Code capability within the camera, but those time codes are not recorded with the DV (Digital Video) option. The Time Code must be connected to the Time Code Reader card to have a TC file created and hence be time coded.

Live Video Backdrop

The Live Video Backdrop allows you to set your streaming live video as the backdrop to your 3D View. To activate this function, right-click in the 3D View and select **Show Video**.

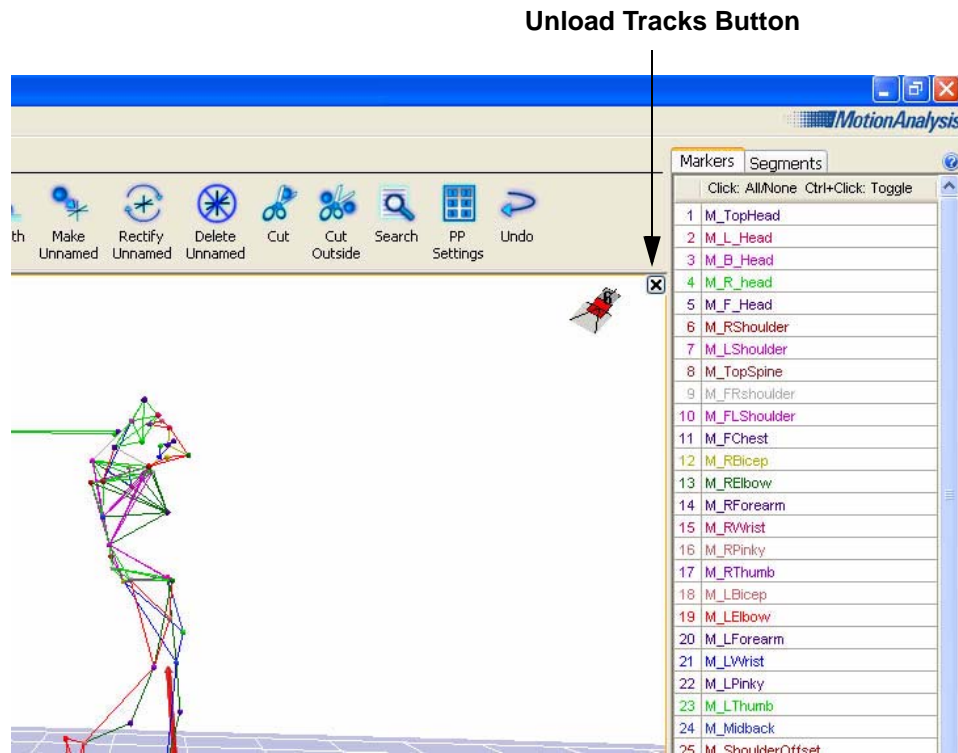
Figure 6-45. Live Video Backdrop



Unload Tracks Button

This button provides a quick method to unload, or not save any changes to, the Tracks files which have been edited in the Post Process mode.

Figure 6-46. Unload Tracks Button



Topic	Page
Getting Started	7-1
Cameras Panel	7-2
Eagle and Hawk Camera Display Codes	7-8
Creating and Clearing Masks	7-11
Going Live	7-11
Adjusting Thresholds	7-12
Analog Panel	7-13

Getting Started

Before using **Cortex**, you must configure your software to match the overall system. The System tab provides tools to do this. The camera settings do not need to be reset before each and every motion capture session but they do need to be reset after changes are made to the cameras.

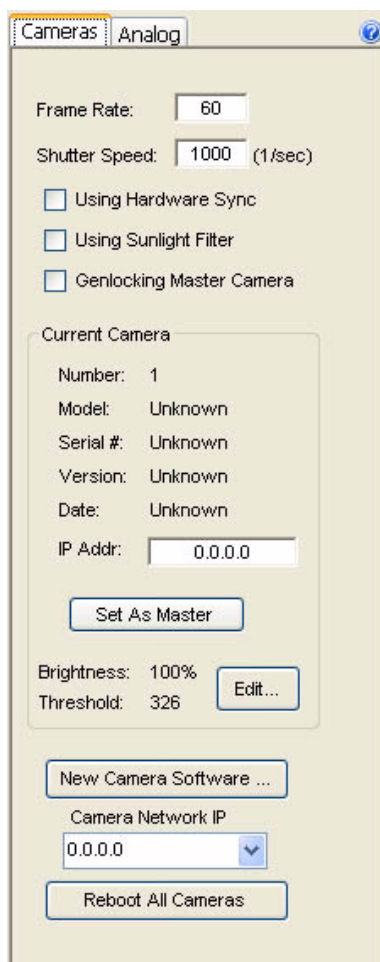
1. Choose **File > Load Project...** from the Menu Bar and load a recent or sample project.
2. Choose **System** from the Mode Buttons.
3. Choose the Cameras panel from the panel buttons if it is not already open.
4. Change the frame rate to the desired level.
5. Leave the Shutter Speed set to the default for normal data collection.
6. Click **Connect to Cameras** on the Real Time Dashboard if your cameras and connections are fully operational.

The system is now ready to go live with the **Run** button.

Cameras Panel

To connect to the cameras successfully, you must have the **[Eagle Support]** item in your **mac_lic.dat** file.

Figure 7-1. Cameras Panel



Frame Rate

Sets the frame rate of the Eagle digital camera to any number ranging from 0.1 to the maximum frame rate. The number does not have to be an integer; it may be set to 59.97, for example.

Note: The maximum supported frame rate for the Eagle digital camera system is 500 Hz. Please contact Motion Analysis Customer Support for information and technical advice for using frame rates higher than 500 Hz.

Shutter Speed

Sets the shutter speed of the Eagle digital camera ranging from 0 to 2000 μ s. This pulse is issued in conjunction with the timing of the strobe (ring light) pulse. There are 1024 different levels of shutter speed control.

Using Hardware Sync

This is selected when you are capturing data at high frame rates (greater than 500 Hz) and the cameras are wired together with the Hardware Sync cable

Using Sunlight Filter

When activated, this feature eliminates large blobs (targets) and one-pixel blobs in the camera hardware caused by typical outside lighting. The tracking parameter “Max Horizontal Lines per Marker” gets set as the max size allowable target in horizontal pixels.

Genlocking Master Camera

This is selected when the master camera is synced to an external video source, either NTSC or PAL. To enable the feature, you must have a license feature installed in your mac_lic.dat license file that looks something like the following line:

[Eagle Genlock] 9c3856f6 782cb125

Please contact support@motionanalysis.com if you need this license item.

To turn it ON, check the box called **Genlocking Master Camera** in the **System > Cameras** panel. When this is done, the Master Camera (which can be any of your Eagle, Eagle-i, Hawk or Hawk-i cameras) must have an analog video signal (black burst or other signal) applied to the camera. This is done using the BNC connector of the 2 meter long Eagle Test Cable that came with the Motion Analysis system. It connects to the master camera using the AUX connector on the back of all the MAC digital cameras. Failure to connect the video signal to the Master camera will show up when you press the Run button. The slave cameras will send data but the Master camera will not.

This feature is available on all Motion Analysis digital cameras and can be set to any multiple of the NTSC or PAL frequencies that your mocap camera will allow. So you can capture at 59.94 (NTSC frequency) or 119.88 (2X NTSC) or 179.82 (3X NTSC) or higher if your motion capture camera will allow it. For Eagle/Eagle-i cameras and NTSC genlock sync, you need to set the camera frame rate to 59.94 Hz (on the Cameras sub-panel). For PAL genlock sync, set the camera frame rate to 50 Hz (or 100 or 150 or 200 or so on). The slave cameras will follow the master camera without any extra wiring.

Current Camera Information

Displays the selected camera number and its corresponding IP address, software version number, along with the date and time the Eagle software was compiled.

Set as Master

Sets the selected camera as a master camera. A master camera generates synchronized pulses to the rest of the cameras within the system so that all camera shutters are opening and closing at the same rate.

Note: Only one master camera can be set for each system.

The master camera is set as follows:

1. In the Real Time Dashboard, select the camera number button of the camera which you would like to set as a master camera.
2. Click on the **Set Master Camera** button.

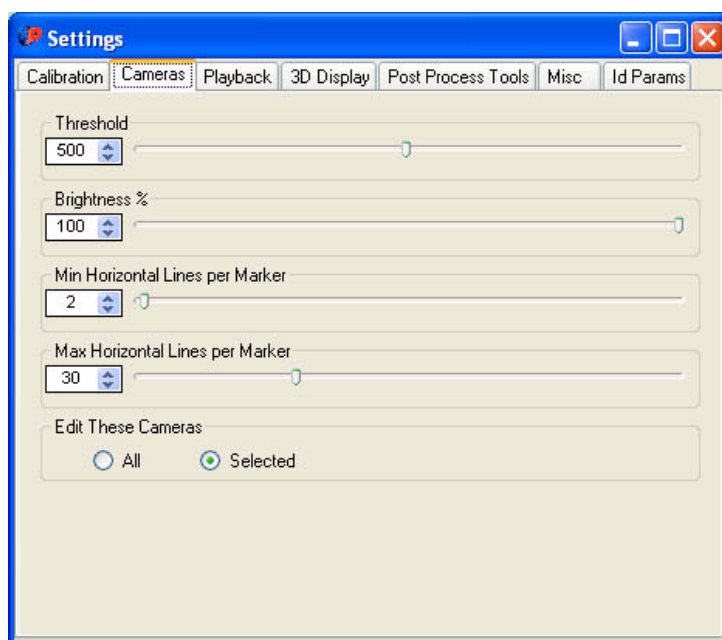
3. If the camera is not turned on or working, select another camera in the Real Time Dashboard, and then press **Set Master Camera** again.
4. If you have an analog sub-system, the A-D sync cable must be connected from the master camera to the A-D Interconnect box. See [Figure B-3 on page B-6](#).

Any camera may be designated as a master camera, but only one at a time.

Edit Button

This button opens the **Tools > Settings > Cameras** interface which features adjustment sliders for Threshold and Brightness settings, Min Horizontal Lines per Marker, and Max Horizontal Lines per Marker.

Figure 7-2. Edit Button (Tools > Settings > Cameras) Interface



Changing the Camera's IP Address

You can change the IP address in this box for any camera at any time. You need to make sure that you do not use duplicate numbers though. It is recommended that you use the same IP address number scheme as used when the cameras are first shipped (10.1.1.xxx). The last three digits should be any number between 1 and 250. In the event that your local area network is set to a 10.1.1.xxx IP scheme, you can also use 10.1.2.xxx for the Eagle camera network (Eagle Host computer, EagleHubs, Eagle cameras, etc.).

Loading New Camera Software...

This allows you to select and upload new software, when available, into the selected camera.

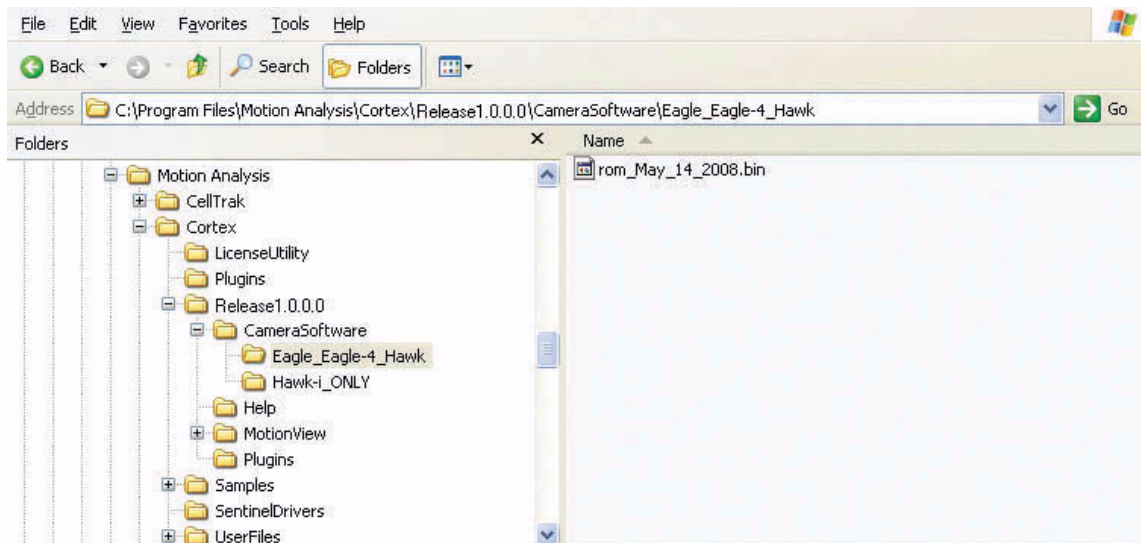
Note: The software loaded into the camera must be of the form **rom_date.bin**. The specific date in the file name may vary.

Note: Hawk-i cameras require a different version of the **rom.bin** software from the other digital cameras. Loading of incorrect software may cause your Hawk-i camera to cease functioning. The **rom.bin** file for the Hawk-i cameras is found in a folder of its own, under the Camera Software directory.

To ensure of the latest software release, the **rom.bin** file will have the date and time of the program in the file name (e.g. **rom_Jun_23_2006.bin**). Install the new software as follows:

1. Obtain the latest **rom_(date).bin** file from Motion Analysis Corporation by means of either an FTP site, e-mail, or disk.
2. Copy the **rom_(date).bin** file into the following directory:
C:\Program Files\Motion Analysis\Cortex\Camera Software

Figure 7-3. Camera Software Directory



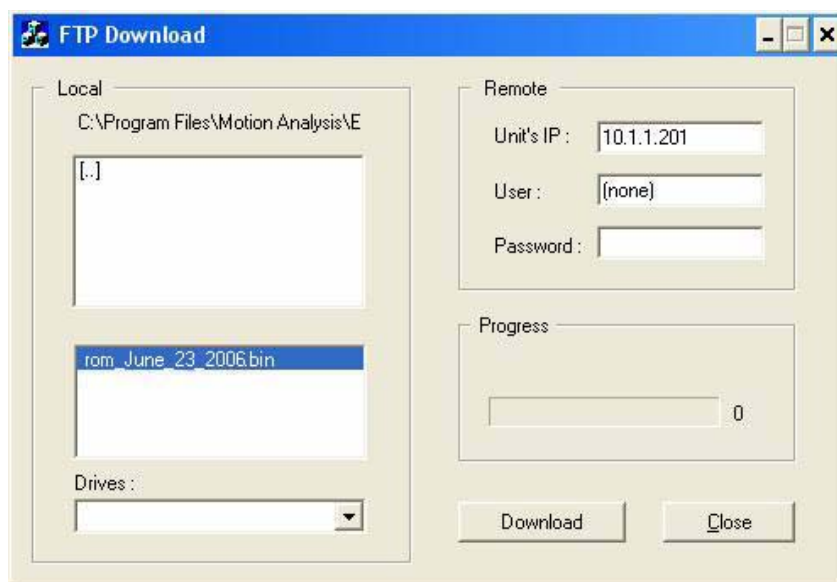
3. Return to the Calibration panel in the **Cortex** user interface.
4. Select the camera (on the Real Time Dashboard) you wish to load the new **rom_(date).bin** software into and click on the **New Camera Software...** button in the Cameras panel.

Figure 7-4. Download New Camera Software Verification Dialog for Eagle and Hawk Cameras



5. Manually type in the Unit's IP address prior to clicking on the **Download** button.
6. Navigate to the **Camera Software** directory with the **[..]** and select **rom_(date).bin**.

Figure 7-5. New Camera Software Interface, Loading the rom.bin File



7. Click **Download** and wait about two minutes for the Writing to Flash operation to finish.

Note: If the message "Send Failed" appears, ignore and press **Download** again.

8. Once the software has been loaded, the cameras may blink oddly. After this, you will need to reboot all the cameras by cycling the power on the EagleHub.
9. Run the **Cortex** software and click on **Connect to Cameras**.
10. Finally, check all cameras for the new software version number.

Note: For Eagle and Hawk Camera users, the camera software with this release is:

\CameraSoftware\Eagle_Eagle 4_Hawk\rom_May_14_2008.bin

For Hawk-i camera users, the camera software is:

\CameraSoftware\Hawk-i_ONLY\rom_Hawki_Mar_03_2008.bin

Do I need to update camera software?

You can continue to use the same Eagle or Eagle-4 or Hawk **rom.bin** software. Reasons to upgrade to the new camera software include:

1. Displays camera numbers greater than 99.
2. Mixed camera environment with all cameras (including Eagle, Hawk, and Raptor).

Old Camera, New Camera Compatibility Issues

Cameras with different revisions of the CPU board installed may exhibit problems with different versions of the rom.bin software.

How to tell which cameras have the OLD CPU board and which cameras have the NEW CPU Board is easy:

- ALL Eagle, Eagle4, and Hawk Cameras with the SILVER backplate (where the connectors are) have the newer CPU boards (CE approved) and require the newer camera software, dated **May 9, 2005** or later.
- Cameras with the BLACK backplate have the older CPU card (non CE approved) can use either the NEWER or the OLDER rom.bin software.

The **rom_Mar_11_2004** software was released with the **EVaRT 4.2** software.

The **rom_May_9_2005.bin** software was released with the **EVaRT 4.4** software.

The **rom_Jan_23_2006.bin** software was released with the **EVaRT 5.0.x** software.

Note: Raptor camera software updates use a different interface which will be available in the next version of this User's Manual

Dedicated Interface for Eagle Cameras

Allows you to input the IP address for the network interface card (NIC) of the host computer. This is the IP address for the NIC that is connected to the Eagle cameras. There must be a dedicated NIC for this purpose. Other connections to local area networks (LAN) must be done on a different NIC to avoid network traffic on the Eagle camera network and to keep the Motion Analysis camera system working properly.

Reboot All Cameras

Reboots the cameras (cycles the power). This is used when changing the camera's IP addresses. Note that the camera and software will not recognize the change in IP address until the camera has been rebooted.

Eagle and Hawk Camera Display Codes

The Eagle and Hawk digital camera displays indicate which mode the cameras are operating. Note that Hawk-i and Eagle-i cameras do not have the display feature.

Table 7-1. Eagle and Hawk Camera Display Codes



Code	Image	Description
Master Camera		<p>This display code, with the active LEDs in the four corners, indicates that the camera has been set as a master camera. The ringlights are ON, which indicates that Connect To Cameras on the Cortex interface is active.</p> <p>Yellow (Red and Green ON) numbering indicates that the camera is in an idle state (powered up but not connected).</p> <p>Red number displays indicate that the camera is either disabled, out of sync, or that there is a hardware problem within the camera.</p>
Standard Camera		<p>This display code has no active LEDs other than those set for the number display LEDs. This indicates a standard operating camera.</p>

Table 7-1. Eagle and Hawk Camera Display Codes






Code	Image	Description
Camera Ready for rom.bin Download		This display code, with the active LEDs in a slash through the display number, indicates that the Eagle camera is ready to accept a new rom_date.bin file. This display code will be go away after the new software has been installed and the camera is rebooted.
Press Download		This display code, with active LEDs in an arrow and rectangle pattern, indicate that the Download button, in the Download FTP window, is ready to be pressed.
Rom.bin Download in Progress		This display code indicates the progress of the download process. The number of activated LEDs will increase as the download process nears completion.

Table 7-1. Eagle and Hawk Camera Display Codes

Code	Image	Description
Rom.bin Download is Complete		
Camera ON, not Connected to Cortex		This display code, with both green and red LEDs activated for the number display, indicates that the camera has been powered ON, but is not connected to Cortex .

Creating and Clearing Masks

Masks are rectangular regions in the 2D View that you designate to receive no marker information. Masks allow you to block out fixed light sources that cannot be physically removed from a camera's view. Masks are created on the 2D Display by clicking and dragging the middle mouse button and appear as hatched regions.

Masks can be cleared by using the right mouse pop-up menu item in the 2D Display and choosing either **Delete Mask** or **Delete All Masks**. Note that you can use masks after the VC data has been collected by disconnecting from the cameras, creating the mask, and then loading or selecting the raw files. This applies to the calibration files (CalSeed and CalWand) as well.

Hardware Masks

For Eagle and Eagle4 cameras, the first 12 masks are hardware masks. Hardware masks are created in the camera, and the marker edges do not get sent to the **Cortex** software. Masks 13+ become software masks: the edge data is sent to the **Cortex** software and ignored (no centroid is calculated). Hardware masking is generally preferred as it reduces the Ethernet traffic and reduces the size of the VC files.

Going Live

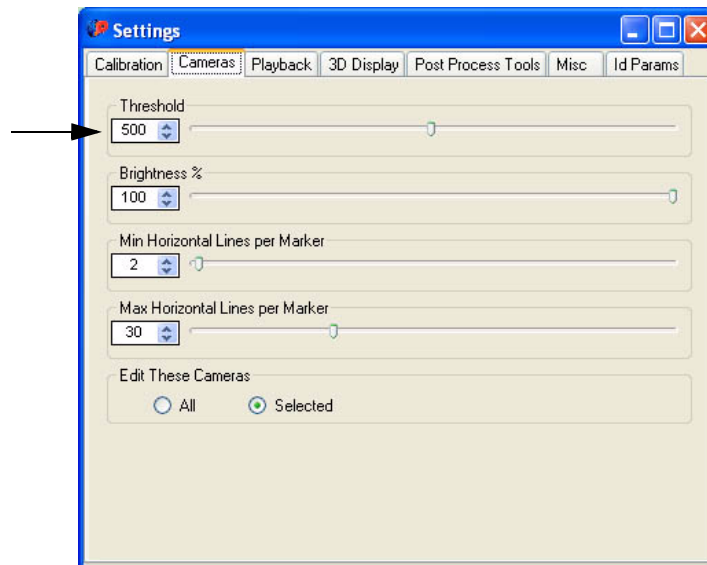
After having configured the software to the system and connected **Cortex** to the cameras:

1. Place the calibration L-frame on the floor in the capture volume. The orientation of the calibration L-frame determines the directions of your global X, Y, and Z axes.
2. Press **F2** on the keyboard to open the 2D Display. The view seen by one or several cameras will be displayed. To select multiple cameras, press **Shift** + click or **Ctrl** + click on any of the camera buttons on the Real Time Dashboard or click **All On**.
3. Click the **Run** button on the Real Time Dashboard.

Adjusting Thresholds

1. Choose **Tools > Settings > Cameras** from the Menu Bar.
2. Slide the Threshold slider on the floating Thresholds control until the markers on the floor appear on the screen.

Figure 7-6. Threshold Slider



3. Select a camera.
4. If you are not seeing any blobs on the screen, choose **Motion Capture** from the Mode Buttons. Then in the Tracking Panel, set the Min. Horizontal Lines per Marker to **2** and Max Horizontal Lines per Marker to **100**.
5. Mask out any unwanted light sources by creating a mask with the middle mouse button held down. Delete masks by clicking on a mask and pressing **Delete** on the keyboard or right-clicking in the 2D Display and selecting **Delete Mask**.
6. Repeat Steps 4 through 5 for all cameras.

Analog Panel

For users who have integrated force plates into their motion capture system, you will need to configure the EMG or other analog source signals for the analog signals to be collected properly. This is done by following these steps.

1. Choose the **System > Analog** panel.

Figure 7-7. Analog Panel

	Name	On	Range
1	F1X	<input checked="" type="checkbox"/>	+/- 10 V
2	F1Y	<input checked="" type="checkbox"/>	+/- 10 V
3	F1Z	<input checked="" type="checkbox"/>	+/- 10 V
4	M1X	<input checked="" type="checkbox"/>	+/- 10 V
5	M1Y	<input checked="" type="checkbox"/>	+/- 10 V
6	M1Z	<input checked="" type="checkbox"/>	+/- 10 V
7	F2X	<input checked="" type="checkbox"/>	+/- 10 V
8	F2Y	<input checked="" type="checkbox"/>	+/- 10 V
9	F2Z	<input checked="" type="checkbox"/>	+/- 10 V
10	M2X	<input checked="" type="checkbox"/>	+/- 10 V
11	M2Y	<input checked="" type="checkbox"/>	+/- 10 V
12	M2Z	<input checked="" type="checkbox"/>	+/- 10 V
13		<input type="checkbox"/>	
14		<input type="checkbox"/>	
15		<input type="checkbox"/>	
16		<input type="checkbox"/>	
17		<input type="checkbox"/>	
18		<input type="checkbox"/>	
19		<input type="checkbox"/>	
20		<input type="checkbox"/>	
21		<input type="checkbox"/>	
22		<input type="checkbox"/>	
23		<input type="checkbox"/>	
24		<input type="checkbox"/>	
25		<input type="checkbox"/>	
26		<input type="checkbox"/>	
27		<input type="checkbox"/>	
28		<input type="checkbox"/>	
29		<input type="checkbox"/>	
30		<input type="checkbox"/>	
31		<input type="checkbox"/>	
32		<input type="checkbox"/>	
33		<input type="checkbox"/>	
34		<input type="checkbox"/>	
35		<input type="checkbox"/>	
36		<input type="checkbox"/>	
37		<input type="checkbox"/>	
38		<input type="checkbox"/>	

Force Plates

☐ Autozero Forces

Analog Sample Rate

Multiple of Frame Rate:

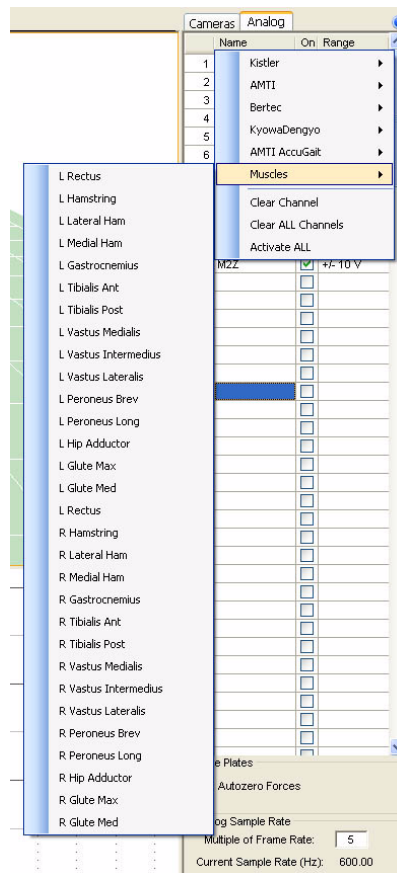
Current Sample Rate (Hz): 600.00

2. To open a list of force plate names, right-click anywhere on the Analog panel grid and choose **Channel Type Names** from the pop-up menu. The built in names include Kistler, AMTI, Bertec and Muscles.

3. To edit a channel's name, left-click in its row in the Name column.
4. Left-click on the arrow that appears in the cell and select a name. Alternatively, you can simply left-click in the cell and type in a name directly.
5. Left-click in the channel's row in the Active column if you wish to make it active. You can also click in the Active column's title cell or right-click on the Analog panel grid and select **Activate All Named Channels**. Both actions activate all named channels.
6. Left-click in the channel's row in the Range column and click on the arrow to select a voltage range. The range must match the output of your analog device.
7. Select the correct sample rate for your system from the Sample per Second drop list at the bottom of the panel.
8. To select EMG muscle names, right-click in the Analog panel and select **Muscles**. For any particular analog channel number, left-click in the Name column and scroll through the drop-down menu for the EMG muscle name you want (see [Figure 7-8](#)).

Note: The Forceplate and EMG muscle names are consistent with the names used in the **Orthotrak** Gait Analysis and KinTrak software from Motion Analysis.

Figure 7-8. EMG Muscle Name Selection



9. To save the entries in the Analog panel, choose **File > Save Project** from the Menu Bar.
10. If you would like to give the file a new name or save it to a different directory, choose **File > Save Project As...**

Shifting Selected Analog Data

This series of functions is used to correct the time-shift-delay in certain telemetered EMG channels. These are introduced by Noraxon EMG units (model 2400 and later), which have a 15ms delay in their signal transmission protocols. As a result, the EMG signals in multi-source (EMG and Forceplates) analog data collected by the Motion Analysis system become non-synchronous with the motion data.

Cortex allows the user to time shift the EMG data in the analog channels using the following steps:

1. Load a Project File
2. Load a Tracks file (.trb/.trc).
3. Select **Data Views > Analog Graphs**.
4. In the Analog Graphs display, right-click and select the **Shift the Data** option.
5. In the Visible Channels dialog, select only the channels you want shifted (typically the EMG data).
6. In the Shift the Data dialog, select the **Selected Channels** option.
7. Set the value for the "Shift the data this number of samples". This can be calculated by the following formula:

$$\# \text{ of samples to shift} = (\text{Analog Sample Rate}) \times (\text{Time Delay})$$

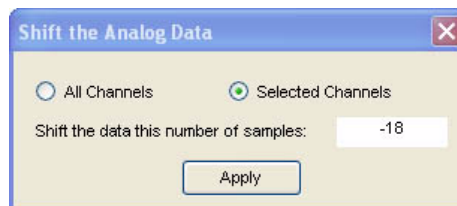
Example

For an analog sampling rate of 1200 samples/second and a time delay of 15 ms (15×10^{-3} seconds) the calculation would be:

$$\# \text{ of samples to shift} = (1200 \text{ samples/sec}) \times (15 \times 10^{-3} \text{ sec}) = 18 \text{ samples}$$

1. To input this frame shift correctly, enter the number as negative value (-18).
 Negative values indicate a shift to the left (decreasing the delay), positive numbers indicate a shift to the right (increasing the delay).
2. To change the entered value from red to black, press **Enter**.
3. Click on **Apply**. When it prompts you with "Would you like to re-write the analog file?", select **Yes**.

Figure 7-9. Shift the Analog Data Dialog Box



Maximum Analog Acquisition Rate

The maximum analog rate is determined by one of the following considerations:

1. The maximum throughput of the National Instruments (NI) A-D product
2. The video sample rate multiplied by 255

The NI USB-6218 A-D unit, which is often supplied with Motion Analysis systems, has a maximum throughput of 250,000 samples/sec (per unit). So two units with 64 channels would have the ability to collect up to 500,000 samples/sec. Either way, it is 250,000/32 channels or about 7500 samples/sec per channel with all channels collecting data. If you cut down to 16 channels, you can have 15,000 samples/sec per channel. You can also connect up to 6 devices and use only 16 channels from each device.

For increased speed, use the faster USB-6259 A-D unit from NI. This unit is 32 channels, 16 bits (same as USB-6218), but has a throughput of 1.25 Million samples/sec or 5-times greater than the throughput of the USB-6218.

The video sample rate multiplied by 255 limitation comes into play mainly if a slow mocap video rate (e.g. 60 Hz) is being used. The max analog rate in this case is 15,300 (60 X 255). Under normal conditions, customers using higher analog rates would also be using higher mocap video rates. With a mocap video rate of 200 Hz, the max analog setting is 200 multiplied by 255 or 51,000 Hz. Note that you can do this with the USB-6218 with fewer channels turned on: $250,000 / 51,000 = 4.9$ (or 4 channels).

Topic	Page
Calibrating Your System	8-1
What is the Square and Wand Calibration?	8-2
Calibrate Panel	8-3
Calibration Files	8-18
Calibration from Previously Collected Files	8-19
Extending the Seed Calibration	8-20
Post Processing Square and Wand Data	8-21

Calibrating Your System

A new calibration must be performed whenever:

- camera positions have changed
- the coordinate system orientation has changed
- the units of measure have changed

It is imperative to complete an accurate calibration in order to collect high quality motion data.

Calibrating your system is a two step process. First, the seed calibration is done by employing the calibration L-frame. The exact positions of these markers must be known. Next, a wand with precisely located markers is waved around throughout the capture volume by somebody wearing no reflective material. Wand calibration ensures that a direct measurement of an object of known size has been made by all cameras throughout the entire capture volume.

This process locates the exact positions of your cameras and accounts for any geometric distortion the camera lenses may have, as well as accurately measuring the camera lens focal-lengths. The importance of this information is so great that a new calibration must be completed if a camera is moved or even accidentally bumped.

What is the Square and Wand Calibration?

In **Cortex**, a four-point square and wand calibration has proven to be very robust and is extremely accurate. This method requires only four markers on the L-frame and three markers on the wand.

Note: If using an L-frame, orient your markers in the same directions as illustrated. Care should also be taken in placing the 4 points on the floor as this determines the global axes and the orientation of the volume displayed in **Cortex**. The points on the 3-point axis must be in a straight line and the spacing of point 2 must be close to $1/3$ of the distance between points 1 and 3.

Calibrate Panel

Figure 8-1. Calibration > Calibrate Panel

Calibrate

☐ Protect Lens Correction

Calibration with L-Frame

☐ Camera Aiming

Details ...

Filename: CalSeed

Calibrate

☐ OK to Overwrite

Calibration with Wand

Duration: 60 (Seconds)

Length: 500

Filename: CalWand

Calibrate

☐ OK to Overwrite

Floor Calibration

Marker Center to Floor: 12.7

Filename: CalFloor

Calibrate

☐ OK to Overwrite

☐ Refine Camera Positions

Property	Value
Camera No.	1
X	-2470.197
Y	2514.597
Z	2939.508
Elevation	-32.12
Azimuth	-132.15
Roll	19.02
Principal Pt. U	620.8273
Principal Pt. V	497.8354
Focal Length	16.81047

Protect Lens Correction

This locks the lens corrections coefficients for all cameras as saved in your project file. Once you have set your lenses’ focus and zoom factor, the lens distortion maps should not change and they need not be calculated with each wand calibration. With **Protect Lens Correction**

checked, the wand calibration will converge more quickly. So if you do not change the lenses, it is a good idea to leave this box checked for all your calibrations AFTER you have completed a good wand coverage and good wand calibration. The results of the successful wand calibration are stored in your project file and your System Calibration file. The System Calibration file can be saved after each wand calibration. It is the default calibration that is used when you launch the **Cortex** software.

For a very accurate calibration, do the following steps:

1. Uncheck **Protect Lens Correction**.
2. Collect a very thorough wand calibration making sure to cover the corners of all the cameras. Select **Run Again** until the numbers stop changing. Accept the results.
3. Check **Protect Lens Correction**.
4. Disconnect from cameras. Select the **WandCal.vc1** file and press the **Calibrate > Calibration > Calibration with Wand Calibrate** button AGAIN. Let it run once and select **Run Again** until the numbers stop changing, then Accept the results.

The above procedure uses the first Wand Calibration to determine the Lens Distortion mappings and uses the second processing of the Wand calibration to refine the calibration. Subsequent wand calibrations can be run like steps 3 and 4 if you do not change the lens settings.

Calibration with L-Frame

Camera Aiming

This optional check-box has two functions:

1. It is used to give a real-time view for positioning the cameras. This is done in conjunction with using the Cam Field-Of-View function (3D Display Show Properties). The cameras are positioned based on the 3D volume specified and you can zoom in and out to see the change in the field-of-view.
2. It is also used to make sure each camera can see all the markers of the calibration L-frame.

Note: If Camera Aiming is checked and Run is selected, the calibration will be recalculated and previous calibration results will be lost.

**Details... Button
(Calibration
Settings Window
Tabs)**

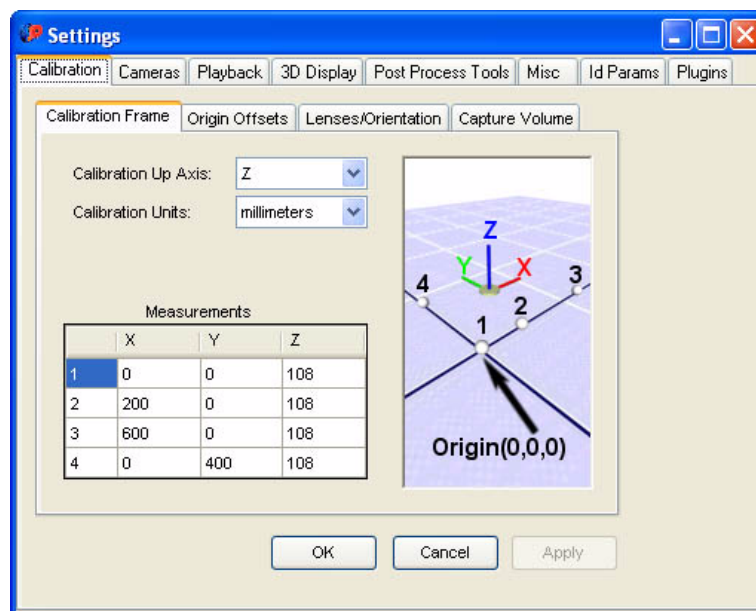
Click the **Details** button, located in the upper-right corner, in the Calibration panel.

The tabs for the calibration settings window, shown in [Figure 8-2 on page 8-5](#), are defined as follows.

Calibration Frame Tab

Where you enter the measurements of the calibration L-frame. You can make your own calibration L-frame by placing four markers on the floor and measuring their locations with a tape measure. Measurements should be within 1 mm. See [Figure 8-2](#) for reference alignment using the Z-up calibration method. Selecting a different calibration up-axis will show the correct view on how to set up your calibration L-frame.

Figure 8-2. Calibration Frame Tab

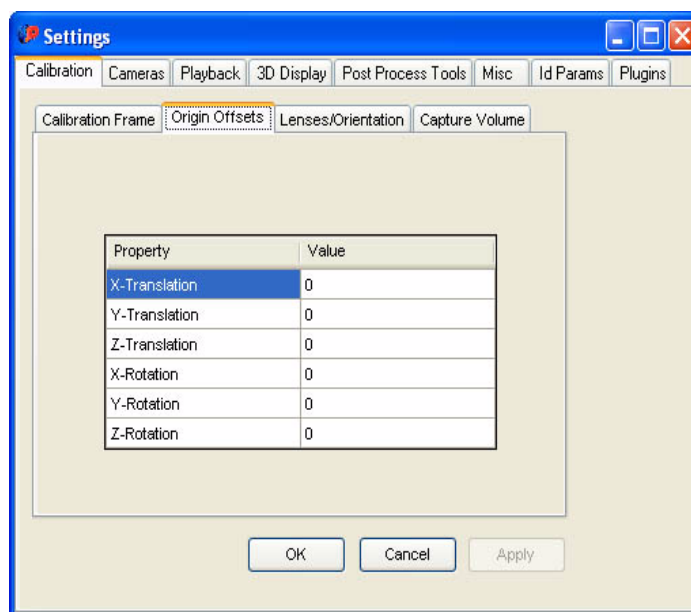


Origin Offsets Tab

This allows for translation and rotation from the origin. The calibration L-frame may then be positioned anywhere in the motion capture area. This is useful for two possible reasons:

1. All cameras do not see the calibration frame, but you want to use it to position the cameras. In this case, you can move the calibration frame to where it can be seen and enter the Origin Offsets measured from the true origin to the (temporary) location of the calibration frame.
2. You want a different location for an origin for any reason.

Figure 8-3. Origin Offsets Tab

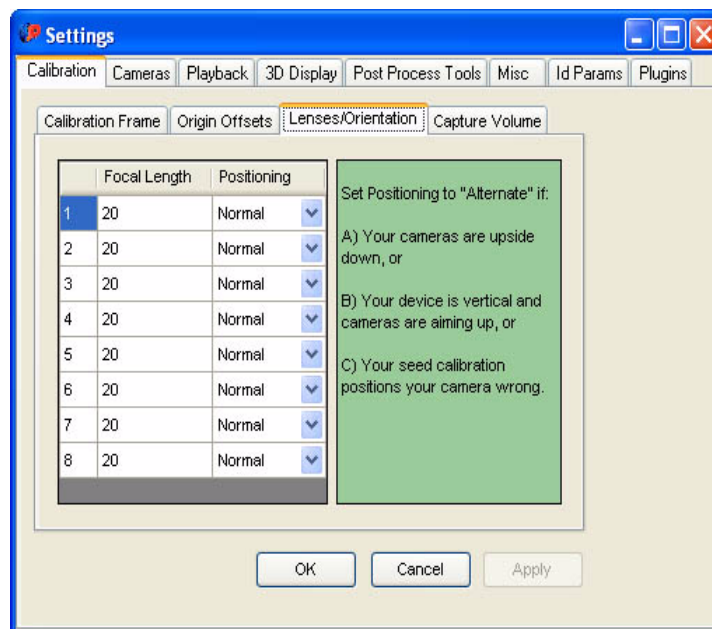


Lenses/Orientation Tab

This tab is used to set the focal length and positions of each camera as used at the start of the wand calibration procedure. If you use the **Camera Aiming** check box in the Calibrate panel to position and orient your camera, the focal length entries should be nominally correct (e.g. 6 for 6 mm lenses). In the Camera Aiming and Calibration with Square functions, the **Show > Show Camera Field of View** cone is determined only by what you put in this table. After the wand calibration, the actual focal length of the lens is calculated exactly and can be stored in your project file.

Note: The **Calibrate Wand** option calculates the actual focal lengths, but does not update the table.

Figure 8-4. Lenses/Orientation Tab

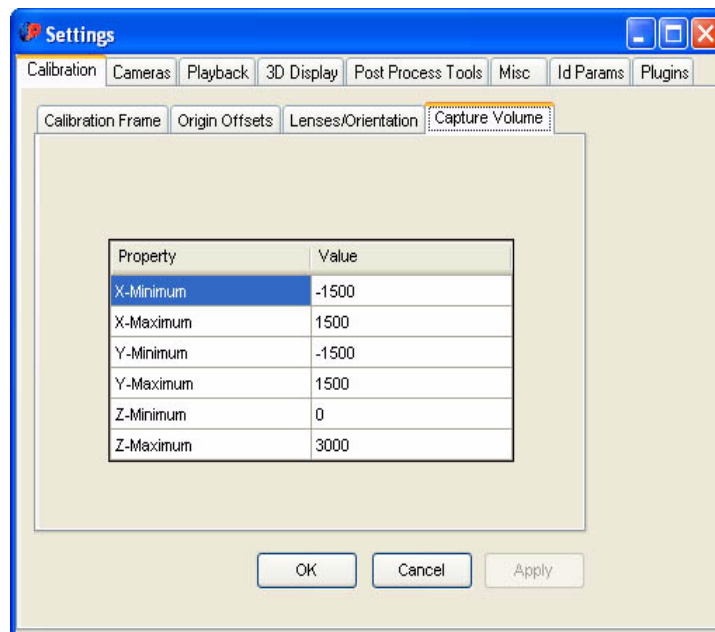


Capture Volume Tab

This displays the rectangular capture volume according to your measurements and helps to provide a visual reference of the volume to the operator. It does not affect the tracked data in any manner. It is for display purposes only in the 3D view.

In the **Plugins > X** panel, the Delete Outside Volume feature will delete any marker data outside of the capture volume.

Figure 8-5. Capture Volume Tab



Square (Seed) Calibration of Cameras

Selecting **Tools > Settings > Calibration > Calibration Frame Tab** (or **Calibration Panel > Details**), will bring up a window as shown in [Figure 8-2 on page 8-5](#). The calibration L-frame should be laid out on the floor exactly as is in this figure. This can be done with four separate loose markers, or it can be done using the wand and a single loose marker, placed at the end of the wand handle. The distances from the origin are measured and are entered into the **Measurements** spaces. Observe the right-hand rule and make sure that you enter the data correctly. In the Z-Up example in [Figure 8-2](#), points 1, 2, and 3 would be at +X, and point 4 would be at +Y coordinates, but adjust accordingly to your Calibration Up Axis. The vertical distances are the distance from the center of the markers (centroid) to the floor. Click on the other tabs and fill in the values accordingly. The Lenses tab should reflect the type of lenses you have in your camera (e.g. 6 mm, 17 mm, 20 mm etc.). The values for the lenses need only be approximations within a factor of two. The actual focal lengths are calculated when you process the wand data. Once you have

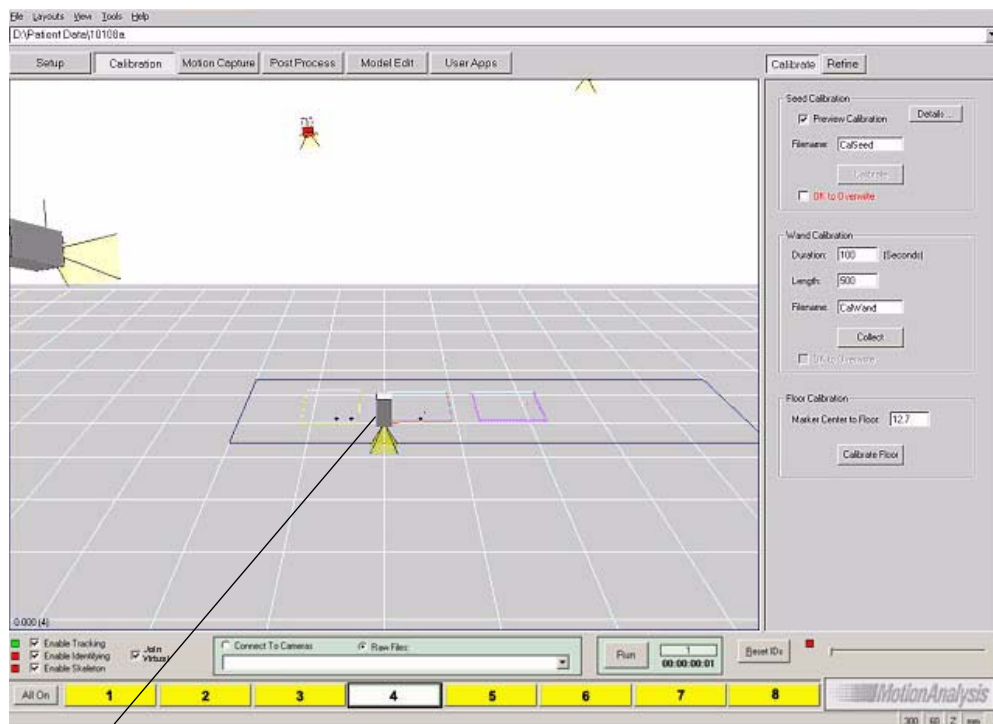
completed filling in the details, press **Apply** and return to the Calibration window in **Cortex**.

Positioning/Aiming Cameras

In the Calibrate panel (under the Calibration tab), check **Camera Aiming** and then press **Run**. The cameras that see four individually defined markers will instantly adjust to their approximate positions in **Cortex** (as in [Figure 8-6](#)). If a camera does not see all four markers or sees more than four markers, it will be displayed at the origin, facing down as in [Figure 8-6](#). This camera will not have a seed calibration, which is acceptable. Refer to [“Extending the Seed Calibration” on page 8-20](#). If the camera is not seeing all of the points, first try one or more of the following steps:

1. Adjust the threshold to see four centroids.
2. Insert Masks to eliminate stray data points.
3. Move the camera position so that it sees four defined markers. In [Figure 8-6](#), a poorly positioned camera will be shown as a camera situated at the origin.
4. If one camera is seen in the exact opposite position in the room, the orientation (up/down) must be changed in the Lenses tab in the Calibration Settings menu. This usually occurs when cameras are tilted more than 90° or mounted upside-down.

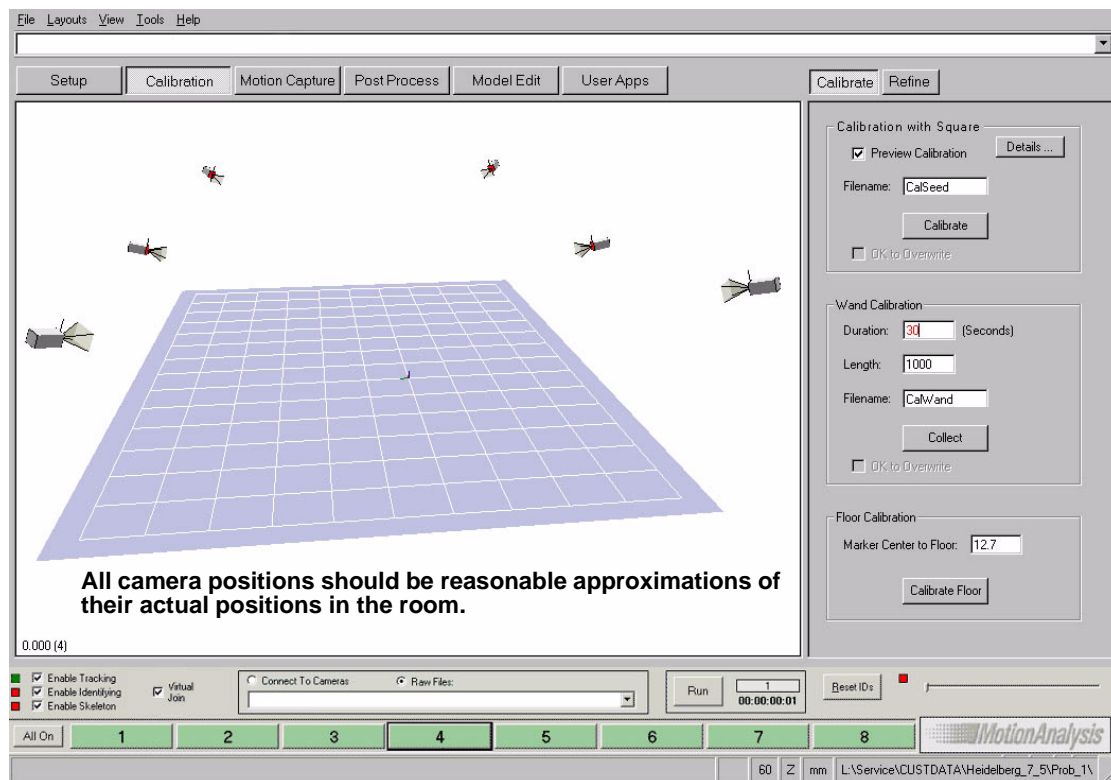
Figure 8-6. Poorly Positioned Camera 4 Results in a Non-Seeded Camera (Camera #4)



Example: The camera does not see the calibration L-frame. Click on the camera to identify it.

Once you have all of your cameras positioned and oriented correctly, press the **Collect and Calibrate** button (see [Figure 8-7](#)). This creates the **CalSeed.vcX** (vc1, vc2... vcN) files and consists of one frame of data stored in your current data folder. Your camera buttons at the bottom should now be yellow in color, indicating that all cameras are seeded (see [Figure 8-8](#)). Fully calibrated cameras show up as Green, but this does not happen until after wand calibration is completed.

Figure 8-7. Properly Seeded Cameras



This completes the square part of the Square and Wand Calibration.

Calibration with Wand

Duration

This function allows the user to set the desired time limit (in seconds) for the wand capture.

Length

This function allows the user to set the length of the wand head (in millimeters), measured from end marker to end marker.

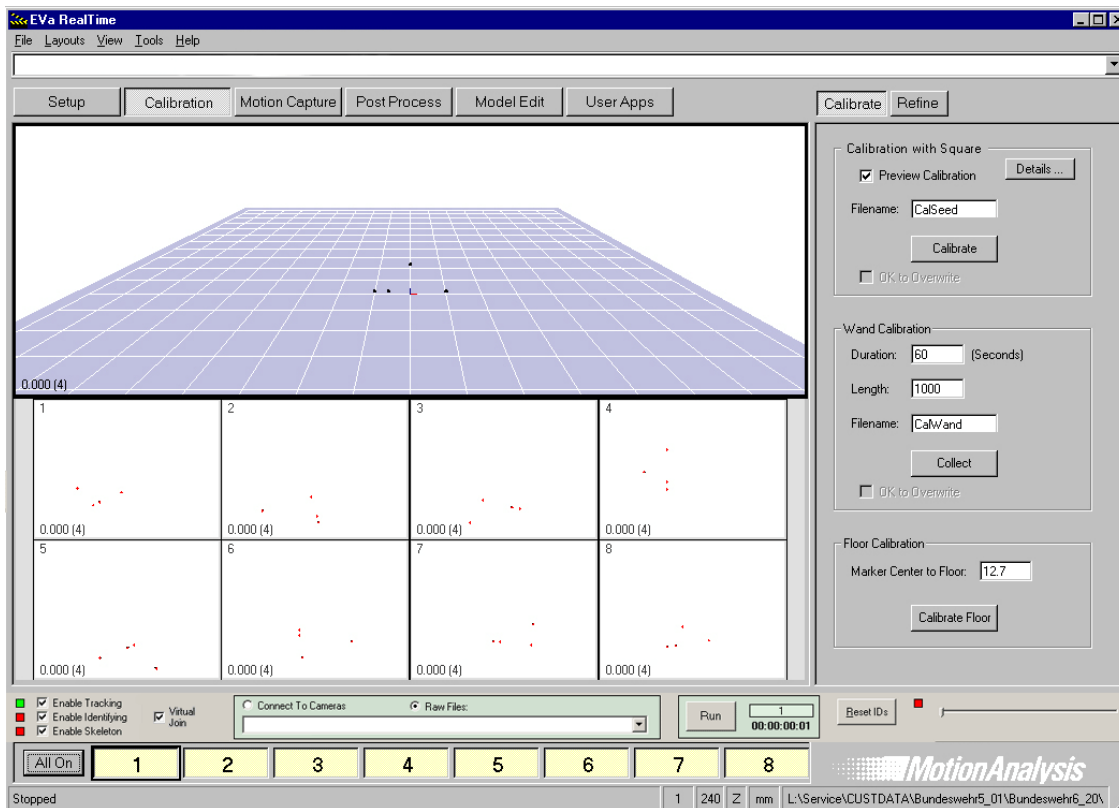
Calibration with Wand Procedure

1. In the Wand Calibration box on the right hand side, set the wand length to your wand size. Make sure that you are using only a three-point wand.
2. Set the duration of the trial. The duration should be sufficiently long enough to wave the wand through most of the volume that you want calibrated. Smaller volumes take less time to complete.
3. Click the **Collect and Calibrate** button and start waving the wand side to side and up and down through the volume. You want to spend about 1/3 of the data collection time with the wand parallel to each of the three X, Y, & Z axes.

Note:

It is recommended that you view the wand movement through the volume at least for the first few times. To do this you must select **Layouts > Top/Bottom**. One window should reflect the 3D view and the area where it is possible to see the wand waving through the volume. The other is the 2D view where the individual camera coverage of the wand in the volume is seen. To show all cameras, press the **All On** button. In the 2D view, the display is automatically smeared in showing the wand data. This shows the volume coverage for each camera. A good wand calibration will fill most of the 2D view for each camera.

Figure 8-8. Seeded Cameras; 3-D and 2-D View (all cameras on)



4. Once the wand calibration duration has been completed, the program starts to determine the volume calibration, and a screen comes up with a series of numbers that decrease as the calibration nears the actual wand length and focal lengths of the camera lenses (see [Figure 8-9](#)).
5. At the bottom of the user interface, a progress bar ticks toward 100%. Once completed, the camera lenses should be very close to what was installed on the camera body (e.g. 6.023 mm for 6 mm lenses). If this is the case, and the wand length is very close (e.g. within 0.10 mm difference for a 500.00 mm wand) to the original wand length, then the calibration is complete. If not, you can continue to click on the Run Again button until either the values stop changing significantly, or the values start getting larger.
6. Press **Save Project**.

Figure 8-9. Wand Processing Status



Extend Seed Button

This button will seed the cameras that were not seeded during the Seed Calibration, based on the wand data. After extending the seed (clicking **Extend Seed**), all the camera buttons should be yellow. Click **Run Again** to complete the wand calibration for all cameras.

Run Again Button

This button continues the refinement of the wand calibration. If the numbers in Figure 8-9 continue to change, click **Run Again** for a more precise calibration.

Accept Button

Click this button if the calibration numbers look sufficient.

Reject Button

Click this button if the calibration numbers are unacceptable. Fix the problem, then select **Run Again**. A typical fix, without having to recollect the data, is that you can create a mask in one of the Raw Video **Cal-Wand.vcX** files.

Stop Button

If you need to stop in the middle of a calibration, or the numbers are not improving during a lengthy calibration click the **Stop** button.

Note: The Stop button is not a cancel button. You can click **Stop** and then **Accept** if you wish to keep the calibration numbers.

Floor Calibration

Floor Calibration Procedure (Optional)

It is necessary to perform a floor calibration if your floor is uneven.

1. After the Seed and the Wand calibration is processed, place several markers on to the floor.
2. Press the **Connect** button and then the **Run** button. You should see the unidentified markers on the floor.
3. Enter the distance from the floor (your $y=0$ plane for Y-up) to the center of the markers (maybe 20 mm) and press the **Collect and Calibrate** button in Floor Calibration.
4. It will tell you how much the calibration origin was moved and rotated with a six number display which stands for the XYZ and yaw, pitch and roll adjustments.

From VC Files

If you are doing this from VC files, you will need to do the following:

1. Select the **CalFloor.vc1** file from **File > Load Raw Files** (with the cameras disconnected).
2. Then press the **Run** button so that you see the unnamed markers on the floor in the 3D view.
3. Then press the **Collect and Calibrate** button in Floor Calibration.

CalFloor.vcX (optional)

You would not normally need this, but it is there to “level the floor” if needed. It is typical to take a single walk cycle and copy and paste it into 100 cycles. If the Calseed device is slightly tilted up or down, this can cause the stick figure to be walking above or below the floor at the ends of the cycles. To correct for this, you can spread 4 or more markers on the floor and press the Calibrate button in the Floor Calibration box.

Note: Make sure no other markers (ghost or otherwise) are visible in the 3D View, as it will tilt your new virtual floor to average them in as well.

Marker Center to Floor

This setting is used to adjust the floor level of the capture volume, based on the size of markers (in mm) used to determine the plane of the volume floor. For example, if 12 mm markers are spread across the volume floor, a Marker Center to Floor value of 6mm (radius of marker) plus any base pad thickness would be entered into this box.

Face Calibration

Refer to [Appendix D, Capturing Facial Motion](#).

Refine Camera Positions Button

After you calibrate the system, you can use the Refine panel to either improve your calibration or to fix the calibration if a camera gets bumped or the cameras have moved slowly over time. When performed well, the Refine Cameras function can greatly improve the accuracy of the system, and will fix a bumped camera, all within 60 seconds. It is one of the most powerful tools in **Cortex**. To use it, check the **Refine Camera Positions** check box press the **Run** button when connected to your cameras. Have a subject move about in the capture volume. Like with wand calibration, you must cover the entire capture volume and field of view for each camera with any marker data. If you get good coverage, then refined calibra-

tion will be very good. If you get only partial coverage, then you the results may be worse than not doing the refine.

Camera Position and Properties

The panel on the bottom of the Calibrate panel displays the details of the selected camera in the 2D view. You can move and rotate the cameras in the 3D View by changing the X, Y, Z coordinates, and the Elevation, Azimuth, and Roll angles. This can be helpful to see if moving the cameras will help with seeing the volume better.

Figure 8-10. Selected Camera Information

Property	Value
Camera No.	1
X	-2470.197
Y	2514.597
Z	2939.508
Elevation	-32.12
Azimuth	-132.15
Roll	19.02
Principal Pt. U	620.8273
Principal Pt. V	497.8354
Focal Length	16.81047

Refine Procedure

1. Connect to cameras (or select a Raw Video file).
2. Select and activate the **Refine Camera Positions** check box.
3. Press the **Run** button on the Real Time Dashboard.
4. Press **All On** on the Real Time Dashboard so that all camera views are displayed on the 2D Display.
5. Right-click on the 2D Display and select **Smear Display**. This will show you how much of each camera's field of view is being filled by the wand over time.
6. Start with the subject in one corner and as soon as the subject starts to move, check the check box for Refine Camera Positions.
7. Have the subject walk around the capture volume filling the entire volume. The subject is acting like a wand calibration. Have the subject walk with both arms slightly out so that all markers are easily identified. Then, have the subject walk the perimeter of the room, spiraling into the middle.
8. As soon as the subject has filled the room and reached the middle, press **Pause**.
9. A table of correction values will appear for all cameras included in the Refine panel. The first three columns are the position changes of the cameras since the original calibration and the second three columns are the rotational changes. Changes of more than 1 mm are often significant and can result in a better calibration.
10. Click **OK** and save as a new project.

11. Press **Run** to start **Cortex** again. You should now see lower residuals and fewer ghost markers.

There are a few things to be concerned about. Since this behaves just like a wand calibration, and if the subject does not fill the volume during the Refine trial, the calibration can be poor and even worse than before the Refine. If the subject spends a lot of time in one area and not much in another, it can also be poor. If the markers are not identified, then there is no information for the Refine, so ensure that the subject is identified. Save the Refine as a new project (e.g. **Refine.prj**) so that if the new calibration is not as good as before, you can go back to the previous project file. The other thing to remember is that the Refine does not guarantee that the scale is exactly maintained. It just optimizes the camera locations to track the markers better. In effect, the scale of the room may change slightly. The reality is that Refining once does not change the scale. It is not recommended to do many Refine Cameras in a row to improve the results, since this may change the scale.

Note: You do not need an identified stick figure to refine, you only need to see 3D unnamed (or named) markers to refine the calibration.

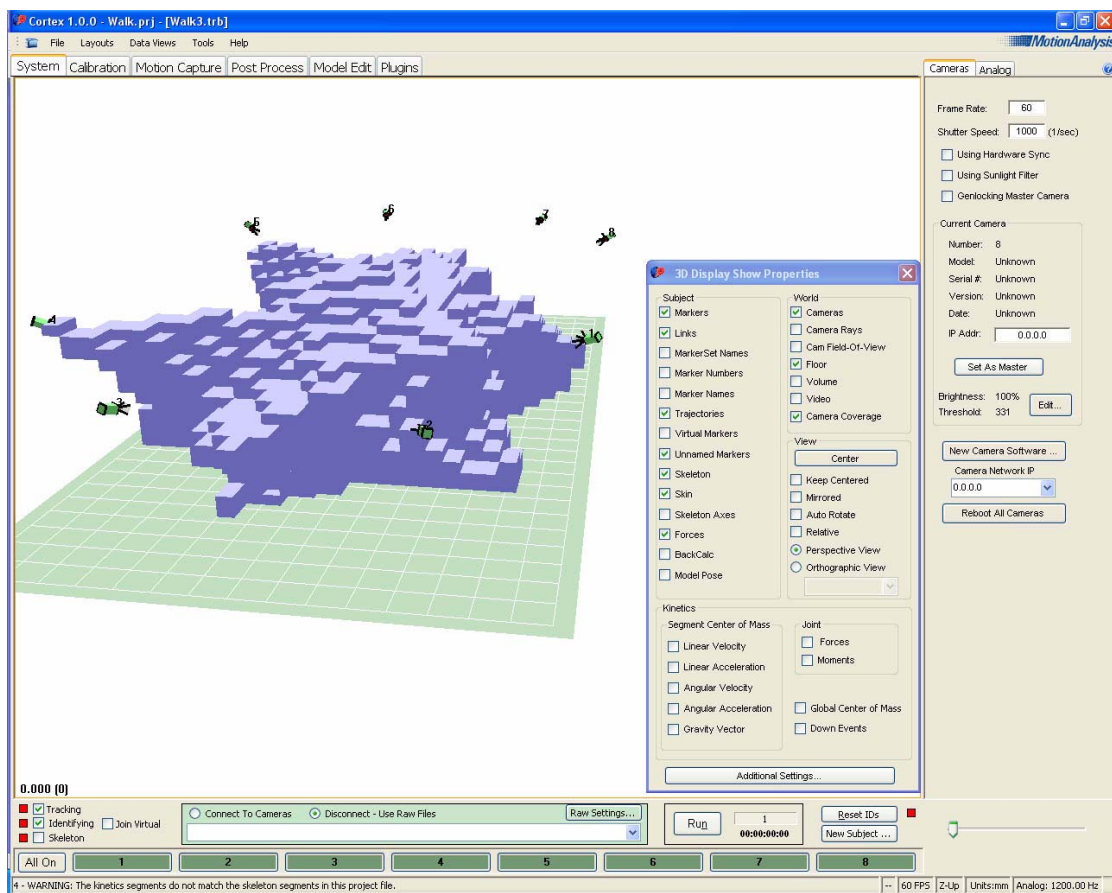
You can use any 3D data points to Refine the calibration. There does not need to be a template or a stick figure. Any data that fills the volume will be sufficient. You can have a marked subject walk in and around the capture volume, with emphasis on the edges. The ray intersection of both the named and un-named markers can be used to refine the calibration. However, if a camera was severely bumped, calibration will be far enough away from ray intersections that the Refine Calibration will not help. In that case, you must redo the L-frame and wand calibration steps.

Show Camera Volume

Show Camera Volume is useful for telling you about how well your cameras are aimed and how much camera overlap you have. To see the camera volume:

1. Right-click in the 3D View and then select the **Camera Coverage** check box in the pop-up. You will need to select the camera number(s) button at the bottom of the Motion Capture panel. Also, you can change the minimum number of cameras that see that part of the volume in the **Tools > Settings > 3D Display > Camera Coverage > Minimum Number of Cameras** setting.
2. Right-click in the 3D view and select **Show Volume**.

Figure 8-11. Show Camera Coverage, Volume



Calibration Files

The following are the types of calibration files generated in your selected capture folder:

Calseed.vcX

Calseed.vcX files (one for each camera) get written when you press the **Calibrate** button in the Calibration with L-Frame box and when you are connected to the cameras. If you are not connected to the cameras, you can use the Disconnect-Use Raw Files item and select the **Calseed.vc1** file to re-process the Calseed files. This is a kind of simulated realtime mode that allows you go back and process the **Calseed.vcX** files and evaluate your data files. When you press the **Calibrate** button in the Calibrate with Square box, it completely removes all of your calibration information and replaces it with the seed or approximate calibration for each camera.

Note: vcX means the set of files that end in vc1, vc2.... vcN if you have N cameras.

Calwand.vcX

Calwand.vcX files get written when you press the **Calibrate** button in the Calibration with Wand box and when you are connected to the cameras. If you are not connected to the cameras, you can process the data in the simulated real time mode as above. The software uses the current system calibration (which is normally the results of the seed calibration, but can be otherwise) and refines the calibration. The calibration includes the exact location and orientation of each camera with respect to the origin, the lens distortion parameters for each camera, and other details about the cameras. At the successful completion of the wand calibration, the software asks if you want to save the system calibration. A **Yes** answer means that a file called **SystemCal.prj** gets written to the system directory.

Other uses for the **SystemCal.prj** file are when you launch **Cortex**, the software automatically reads the **SystemCal.prj** file and when you exit the **Cortex** program, it automatically writes the **SystemCal.prj** file into the system folder. The intent is so that you can launch the **Cortex** software and it will remember its last good calibration without having to load any files. If you load a PRJ file or load a CAL file, it will overwrite the calibration information in memory with the contents of the PRJ or CAL file. Both contain calibration information, but the PRJ file also has the marker set information and template information.

TrialN.cal

It is a good practice to use **TrialN.cal** for every capture you make. Every time you collect a trial in the **Motion Capture > Output** panel, the system writes out the current calibration to a file that has the same name as your trial name, but with a **.cal** extension. This is normally not needed, but will allow you to load up the calibration at the time of the capture with the **File > Load Calibration...** menu item. If you changed the calibration for some reason and you know you were calibrated when the trial was collected, you can load up that as the current calibration in the software at a later time.

Calibration from Previously Collected Files

This section describes how to simulate a calibration using previously recorded data. A simulated collection of **Cortex** square or wand data is done the same way that you can simulate tracking Raw VC data. Just follow these steps:

1. Disconnect from your cameras.
2. Select **Raw Files** on the Real Time Dashboard.
3. Load the **CalSeed.vc1** file. At this point, you can mask out any extraneous data points if necessary.
4. Press the **Run** button on the Real Time Dashboard.
5. In the Calibrate panel, press **Calibrate** in the Calibration with Square box.
6. The cameras buttons on the Real Time Dashboard should turn Yellow which indicates calibration is square, but not wand calibrated. White means not square. Green means fully calibrated.
7. Select **Raw Files** on the Real Time Dashboard.
8. Load the **CalWand.vc1** file. At this point, you can mask out any extra data points that might be causing problems. This feature allows you to utilize wand data sets if there are extra markers in the field of view of one or more cameras.
9. In the Calibrate panel, press **Calibrate** in the Calibration with Wand box. You will see the wand results, see the cameras move to their final place, and see the measured focal distances.

Extending the Seed Calibration

If one or more cameras are not seeing the four-point calibration L-frame device for any reason, they have not been properly calibrated and it shows up as a white camera button. The camera is shown on the floor at the origin point down, as shown in [Figure 8-6 on page 8-9](#). This will result when you have a large capture volume and only some cameras see the calibration L-frame or a camera threshold may be set incorrectly. This is not a problem. You can use the wand data to get the camera calibration seeded, then process the wand data again so the camera (or cameras) get both square and wand processing. The steps for this are as follows:

1. Calibrate using the calibration L-frame as described in “[Calibration Frame Tab](#)” on [page 8-5](#). Cameras that are Yellow are seeded. Those that remain White are unseeded and show up on the floor, at the origin pointing down.
2. Process the wand data. The cameras that saw the calibration L-frame will show as Green camera buttons; the unseeded cameras remain White.
3. Extend the calibration seed by clicking **Extend Seed**. This will then seed those cameras previously unseeded.
4. Click on **Run Again**. This runs the wand data again for all cameras. After this, all cameras should be Green (calibrated).

Figure 8-12. Extend Seed Calibration Button



Extend Seed Button

Note: After you have clicked on **Extend Seed**, check that the residual value for each camera is at a reasonable level. If all cameras do not eventually seed, you will need to check your Wand coverage.

Post Processing Square and Wand Data

It is possible to process the square and wand data after the data has been collected. This is helpful if you did not have the time to process it during the capture or if you lost the project file that contains the calibration information. The steps are as follows:

Note: You must have the **CalSeed.vcX** and the **CalWand.vcX** files from the data capture session.

1. Select **Raw Files** in the Real Time Dashboard and then select **CalSeed.vc1**. Select your CalSeed data set in the data capture folder.

Note: You can create a mask to eliminate unwanted markers or reflections. Re-select the **CalSeed.vc1** file again and press the **Calibrate** button in the Calibration with Square section.

2. In the Calibration with Square section of the Calibration panel, click **Calibrate**.
3. Select **Raw Files** in the Real Time Dashboard and then select **CalWand.vc1**. Select your CalWand data set in the data capture folder.

Note: You can create a mask to eliminate unwanted markers or reflections. Re-select the **CalSeed.vc1** file again and press the **Calibrate** button in the Calibration with Square section.

4. In the Calibration with Wand section of the Calibration panel, click **Calibrate**.
5. Complete the wand calibration as described in [“Calibration with Wand” on page 8-11](#).

At this point you now have your cameras calibrated and you may proceed with your data collection.

Topic	Page
Overview	9-1
Tracking Panel	9-2
Building a Template from the Range of Motion Trial	9-5
Multiple Tracking Objects	9-7
Objects Panel	9-8
Output Panel	9-12
Recording Data	9-14
Tracking Strategies and Tips	9-15

Overview

Motion Capture is the mode where you will spend most of your time during a recording session. In this mode you can:

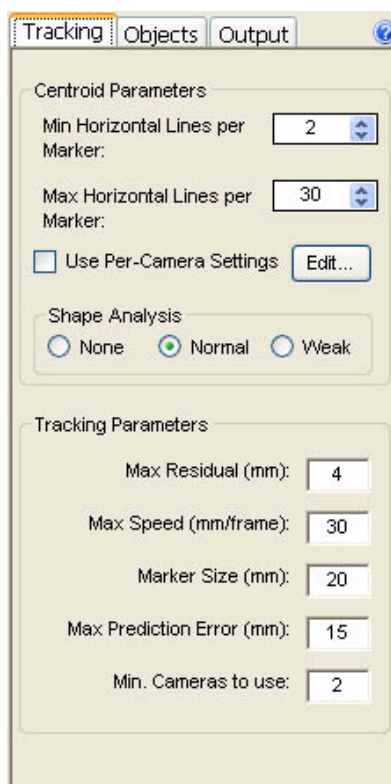
- Create and improve a template
- Set the tracking parameters
- Save data in a variety of file formats

These functions are described in this chapter.

There are a few preliminary steps that must be taken before starting a successful motion capture session. Tracking parameters tuned to your system must be set. Names must be assigned to the markers that will be used. These names constitute a marker set and building this set is actually done using the Model Edit tools discussed in [Chapter 11, Model Edit Panel](#). A template specific to the markers in use must be created. A template describes the minimum and maximum distances that separate linked markers, such as the distance between the right elbow and the right wrist. Templates are created using Motion Capture tools and Post Process tools described in [Chapter 10, Post Processing Panel](#). Once these steps are completed, you are ready to begin a motion capture session.

Tracking Panel

Figure 9-1. Tracking Parameters



The Tracking panel allows the user to set the key parameters that are used when acquiring and tracking data. [Figure 9-1](#) is an example of the parameters entered for a typical setup. These settings are all saved in the **project** (.prj) file. These parameters fall into three categories:

Centroid Parameters

Centroid Parameters control the minimum and maximum number of video lines that are permitted for marker images. If an image size falls outside these limits, no centroid will be calculated for it and it can never become a marker image.

Min. Horizontal Lines per Marker

Sets the minimum number of scan lines a marker must occupy on the camera's sensor for it to qualify as a marker. The value of the parameter entered is dependent on the size of the markers and the distance the camera is away from the markers. A typical value for a 1 inch marker would be **2**. For Eagle-4 cameras, these values will generally double to 4 lines per marker.

Max Horizontal Lines Per Marker

Sets the maximum number of scan lines a marker must occupy on the camera's sensor for it to qualify as a marker. Again, the value of the pa-

	parameter entered is dependent on the size of the markers and the distance the camera is away from the markers. A typical value for a 1 inch marker would be 50 .
Use Per-Camera Settings	This check box provides a quick way for a user adjust the Threshold and Brightness settings for individual cameras, as set in the Tools > Settings Cameras tab.
Shape Analysis	Filters out the centroids of blobs that are not round (e.g. a marker is partially obscured, or two markers have merged).
Tracking Parameters	Tracking parameters are used when correlating the images from several cameras to establish marker coordinates in three dimensions.
Max Residual (mm)	Is the maximum average error when rays from several cameras are combined to establish the coordinates of one marker. If the residual exceeds this amount, it is assumed that these rays are not close enough together to be seeing the same marker. This parameter value should not be less than 4 times, and no greater than 8 times, the average residual value. The average residual value is found in the lower-left corner of the screen when the cameras are running. A typical parameter value is 5 mm .
Max Target Speed (mm/frame)	Sets a speed limit on the markers. A marker's track is eliminated when it surpasses this value. When tracking the tip of a golf club or other object with fast moving markers, it is possible that this value will need to be increased. A typical parameter value is 100 mm/frame .
Marker Size (mm)	Limits the size of the markers so that higher residual cameras do not see more than one centroid for the same marker. This parameter should be set to the physical size of the markers in use (25.4 millimeters = 1 inch). This parameter will also set the size of the markers that appear on the 3D View.
Max Prediction Error (mm)	Is is used to identify a marker in the next frame. While the software is tracking a marker, it is assumed that it will not deviate by more than this amount along its path. Otherwise the marker will not be identified in the frame. A typical parameter value is 30 mm .
Min. Cameras To Use	Tells the software what the minimum number of camera's rays are required to triangulate (track) a marker during a frame. Some users will benefit by setting this value to 3 if spurious data points (ghost markers) are seen in the motion capture sequence.
ICorrectly Identifying Markers Automatically: Motion Capture Mode	<p>In the Motion Capture interface, markers are correctly identified if:</p> <ol style="list-style-type: none"> 1. The current "dynamic template" is good and the data fits into it. 2. All markers are present. 3. The Identifying check-box has been activated. <p>The software keeps a separate dynamic template, which is used during Real Time or simulated Real Time tracking and only applies in the Motion Capture interface. It starts with the user created Template (from the</p>

Create Template button) in the current project and dynamically adjusts itself as the markers are seen to stretch outside of the limits set in the current Template. This is useful when you are using the **Cortex** system for live performances. If markers are all identified, right or wrong, the dynamic template is updated for that frame. Pressing the **Reset IDs** button in the Real Time Dashboard resets the Dynamic Template back to its original, user created Template. This is to be done if the identification gets mixed up for any reason. It is a likely result of the template being stretched too much, perhaps after a mis-identification of markers. This causes the dynamic template to perform poorly and it needs to be reset back to the original user created template.

For the Reset IDs function to be successful, all of the markers used when creating the template must be present. When the markers are un-identified, the software keeps looking in the current dynamic template to identify the markers. The software will also continue to identify the markers whose history it knows about, so you can see frames where some markers are correctly identified and other markers are shown as the black unnamed marker crosses. When about 1/3 or more of the markers become un-identified, the software tries to apply the dynamic template over the entire marker set to re-identify the markers. All of the markers in the marker set must be present before they can be identified.

Dynamic Template Stretch Limits

These are parameters that affect how much stretch the dynamic template is allowed to change, which are set with in the **Motion Capture >Tracking > Identifying Parameters: Linkage Stretch Parameters: To Reconsider (std. dev.) and Max. Acceptable (std.dev)**. This does not apply to the Post Processing Template Rectify feature, but only to the Motion Capture mode of tracking.

To Reconsider

To Reconsider is a unit-less measure of linkage stretch checks which the current frame marker identifies against the current dynamic template. The dynamic template is a measure of the minimum and maximum of each of the linkages, which is updated as the person moves about (and the linkage lengths change in time). If any of the linkages are stretched beyond their limits, the identities of markers at both ends of those linkages are changed to "Unnamed". The limiting factor is taken as multiples of the standard deviation of the linkage length. A typical number for a tight setting is 7 and for a loose setting is 12. A bigger number will allow the template to grow more quickly, but can cause mis-identifications. A smaller number may keep the software from identifying the markers correctly.

Max Acceptable

This is a unit-less measure of linkage stretch that is applied to all unnamed markers in the current frame in an attempt to find proper identities for them. It is applied after any linkages were deemed to have stretched excessively. This number is usually 2 or 3 less than the To Reconsider value.

Building a Template from the Range of Motion Trial

A template tells the software what the minimum and maximum distances are that can exist between markers of a relatively fixed relation. It is necessary to allow the software to identify each marker in each frame. Template information is saved in the project (.prj) file.

Before a template can be created, a marker set that will apply to the subject being captured must exist. If such a marker set does not exist, it must be built using the Model Edit tools described [Chapter 11, Model Edit Panel](#). Once an appropriate marker set exists, follow these steps to create a template.

1. Choose **Motion Capture** from the Mode Buttons.
2. Choose **Output** from the panel buttons.
3. Check the **Tracked ASCII (TRC) or Tracked Binary (TRB)** checkbox on the Output panel.
4. Type a file name in the name box and press **Enter**.
5. Set the Duration (seconds) between **10** and **20**.
6. Start collecting the range of motion data of the subject by having the person stand in the middle of the capture volume with arms extended, palms parallel to the floor with thumbs facing forward, and all markers in full view.
7. Click **Record** on the Output panel.
8. The subject must stay in an initial frozen position for two or three seconds.
9. After standing frozen in this initial position for up to five seconds, the person must move through a complete range of motion by moving and twisting, ensuring that each linkage exhibits the full extent of stretch that will be experienced during subsequent motion capture sessions. Exaggerated motion must be avoided and all markers should remain in full view. This step should not require more than fifteen seconds.
10. After ten seconds passes from the moment **Record** was clicked, the system will automatically stop collecting and tracking marker data.

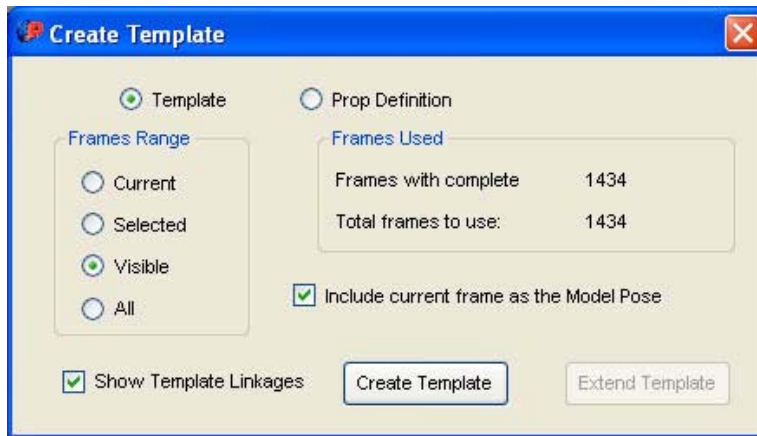
At this point, a Tracked ASCII (TRC) or Tracked Binary (TRB) file has been generated in the current directory and is ready for editing. Next, the markers must be hand identified according to the marker list built for the subject's marker set (assuming that the marker list was already defined prior to the motion capture).

11. Choose **Post Process** from the Mode Buttons.
12. Click **Quick ID** and identify the unnamed markers according to the conventions described in [Appendix C, Marker Sets](#).
13. Click **Rectify**. This applies the naming convention across all the frames of data.
14. Manually cleanup and identify all tracks in this range of the motion file. The template should be defined as at least 75% of the visible frames selected.
15. Select **Template**.
16. Click **Create Template**.
17. Select the appropriate Frames Range:

- **Current**—the current displayed frame
- **Selected**—frames highlighted in blue, low to high in dashboard
- **Visible**—what is displayed across the screen, as a function of the time zoom
- **All Frames**—all frames

18. Click **Create Template**.

Figure 9-2. Create Template Dialog



Prop Definition

Selecting Prop Definition creates a **<projectname>.prop** file that is a rigid body measurement of the object. This .prop file can then be selected as one of the "Additional Tracking Objects" in the **Motion Capture > Objects** panel.

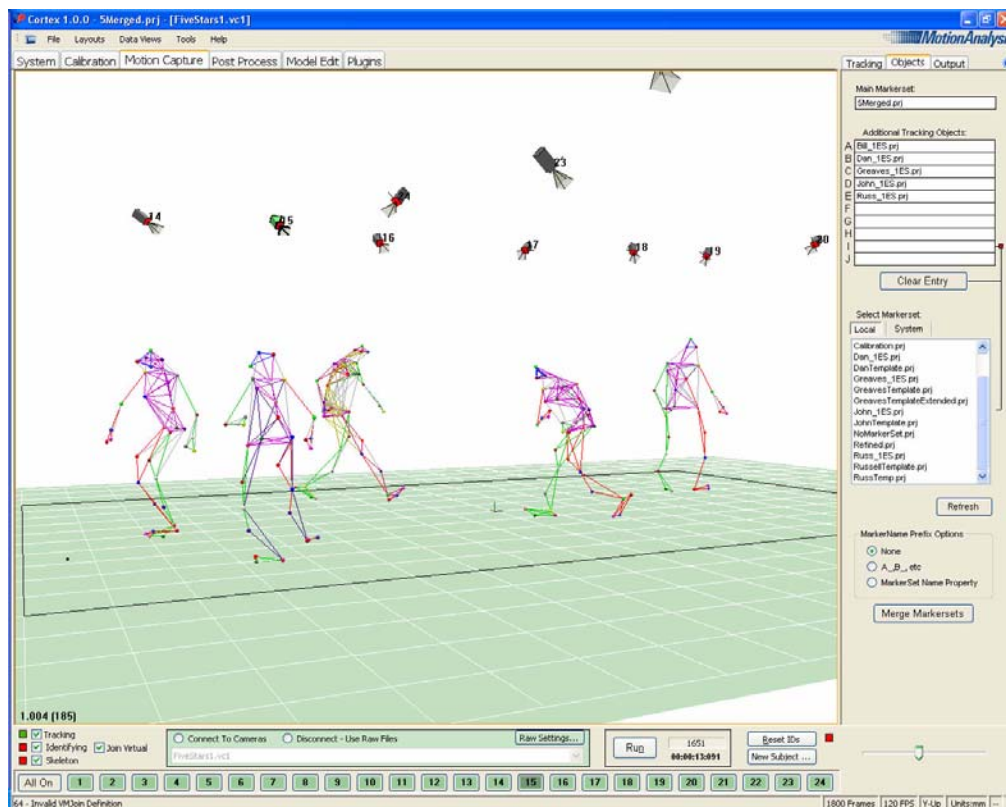
Multiple Tracking Objects

The problem with tracking many people and props (all Tracking Objects) is that you need a unique project file and template for every combination of things you want to track. The project file has to have exactly the right number and names for the combination of markers that you want to use. You then create a template for these combinations and load that project when that combination of objects is to be tracked.

With the Multiple Tracking Objects (MTO) architecture, you need to have one project and template for each object. You record the range of motion for each object separately and create a template for each object and save it in each object's **.prj** file.

For example, when Subject1 and Subject2 and the Subject3 are to be brought into the volume to track, you select the Tracking Objects **Subject1.prj**, **Subject2.prj**, and **Subject3.prj** files. The base project file, entered into the **Main Marker Set:** text box, that you load can be a "Calibration Only" project file (recommended for a higher number of objects that are entering and exiting the data set) or a project file with one marker set. You can then place projects with multiple, pre-defined subjects in the **Additional Tracking Objects:** text boxes. These text boxes can hold up to 5 additional objects. For more information, see the [Objects Panel](#) section on the following page.

Figure 9-3. Multiple Tracking Objects—5 Person Capture

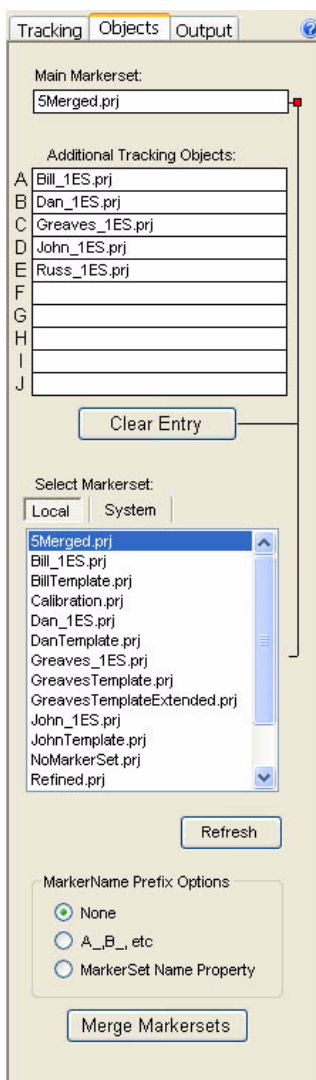


Objects Panel

The Objects panel sets the main marker set and allows you to work with multiple tracking objects (MTOs) while tracking, and bring them into your motion capture data set. A good example of this would be bringing in a second dancer in a dance routine, or a prop such as a golf club when analyzing a golfer's swing.

Under any condition, it is easier for the software to identify a marker set that has greater asymmetry. You may change the order of the additional tracking objects, as listed in the Objects panel to vary the results of your data set. It has been found that the more asymmetrical of your multiple tracking objects should be put at the top of the Additional Tracking Objects list.

Figure 9-4. Objects Panel



Main Marker Set	Allows you to choose which object data set is your primary data set in the tracking session.
Additional Tracking Objects	Allows you to add multiple tracking objects (i.e. people, props, etc.) to your tracking session.
Clear Entry	Clears the selected Additional Tracking Object cell (A through J).
Select Marker Set	<i>Local</i>

Lists and manages which marker sets are to be used in the project tracking session (as set in the Additional Tracking Objects section).

System

System Objects is a feature used to help track objects that are known to be rigid objects. The advantage in using this feature over using a standard objects with a Template is that rigid objects can be more robustly tracked and can have several markers define the 6 DOF Rigid Object coordinate system. This is useful for tracking and identifying props such as swords, baseball bats, and other sport accessories. To use this feature:

1. Create a project file with just the marker names of the prop. Each prop should have its own marker set and associated project file. There must be three or more markers for rigid bodies to work; five or more markers is recommended. Marker names do not matter when they become tracked as Rigid Objects. Linkages are not needed as Rigid Objects have implicit linkages between all markers on the object. You can create linkages to help you visualize the prop when it is tracked if you wish but when a rigid object is identified as such, both the markers and the linkages are shown as a purple color.
2. Create a Prop file. Record some number of frames of the prop (maybe 2-5 seconds), and then load the Tracks into Post Processing. Identify the markers and click the **Create Template** button. Select **Rigid Object** as the template type. The Create Template changes words to "Create Prop File". A message pops up telling you that you created a prop file in your current project folder. A second message informs you that "The type will become rigid when selected on the Objects panel and that a Rigid Object Template has been created.
3. Using the prop file in you local folder, the prop file can be selected as any Tracking Object in the **Motion Capture > Objects** panel. Props can be selected as normal tracking objects and they take up one of the tracking object slots. Currently the coordinate system of the prop is displayed in the Motion Capture 3D View. When you record a trb or trc file, the XYZ coordinates of the prop are also recorded. Visually, you can see the coordinate system of the Rigid Object defined. The coordinate system displayed has the origin at the center of mass of the defining markers and the directions of the XYZ axes are defined parallel to each of the calibration coordinate system axes as defined on frame 1 of the capture.
4. Create a global System Objects folder. If you want to build a library of props, you can create a folder under the launch folder of **Cortex** (next to the Sounds folder) and name it SystemObjects (note: no

space between System and Objects). Copy any props you want to use into the SystemObjects folder. You will need to quit and relaunch **Cortex** and then you will see a new section at the bottom of the **Motion Capture > Objects** panel with the heading "System Objects." You can select prop files from that list to go into any of the "Additional Tracking Objects" slots.

5. Order the objects in the Tracking Objects panel. The priority for identifying objects is A-B-C-D-E and the Main Marker set is identified last. You will want to put the props at the beginning of the list into the top slots. Done this way, it is less likely for the props to be embedded into one of the other human objects.

Marker Name Prefix Options

This provides the option to add prefix designations to marker names for better organization when merging multiple marker sets.

The A_,B_,etc option will place an alphabetic prefix and underscore at the front of the marker names. This will only apply to those markers listed in the Additional Tracking Objects (A through J).

The Marker Set Name Property option will put the project name in front of the marker names. This will only apply to those markers listed in the Additional Tracking Objects (A through J)

Merge Marker Sets

Merges the marker sets of the MTOs to become one merged project.

Advanced Example Data Set

This example data set, found in **C:\Program Files\Motion Analysis\Cortex\Samples\Five Person Tracking** shows a five person capture of 1200 frames of data. Each person has their own project file which defines the marker set template for that person.

The TRB file creation process was done as follows:

1. The **FiveStars1.trb** was generated from the VC files. All markers are unnamed. This was done while using the **All_Five.prj** project file.
2. For each person:
 - a. Load their project file (for example, load the **GreavesTemplate.prj** file).

Note: You may need to increase the number of cameras in the **Tools > Settings > Misc** tab.

- b. Read in the Unnamed marker file (in this case it is the file **FivesStars.trb**).
- c. Saved out to a new TRB file (in this case **FiveStarsGreaves.trb**).
- d. Tracked the data by finding frames where the template identified the markers and used rectify to identify all the markers over all frames. Filled in data where necessary. In the Greaves example go to frame 1, select Template ID, select all frames, and then select Rectify. This will ID all the markers over all the frames.

- e. When fully tracked, deleted all unnamed markers and saved the result to a TRC file.
- f. Used Virtual Marker Join to reconstruct missing markers and saved this back out to the TRC file.
- g. Used the final TRC file to create skeleton for import into Maya.

From the above procedure you can see how the TRB files were used to track visible markers as much as possible. Then all marker reconstruction was done using TRC files.

You can see the difference by loading, for example, the **JohnTemplate.prj** and then loading the **FiveStars1_John.trb** and comparing that to what you see when you load **FiveStars1_John.trc**.

Note: It is important to keep the project files in a particular order for each time the MTO function is performed for a given data set.

Output Panel

The Output panel is used when the motion capture recording is initiated. It is set by the following procedure:

1. Choose the output file type (at least 1).
2. Enter a file name.
3. Set the file length (in time).
4. If necessary, set the external trigger mechanism.

Figure 9-5. Output Panel

The screenshot shows the 'Output' tab of the Cortex 1.0 software interface. The panel is divided into several sections:

- Output files:** A group box containing five unchecked checkboxes: 'Raw video (.vc*)', 'Analog (.anb)', 'ColorVideo (.avi)', 'Tracked ASCII (.trc)', and 'Tracked binary (.trb)'.
- Settings:** A group box containing:
 - A 'Name' text field with the value 'Walk'.
 - A 'Trial #' spinner box set to '3' and an 'Auto-increment' checkbox that is checked.
 - A 'Duration (seconds)' text field set to '60'.
- Buttons and Triggers:** A 'RECORD' button, a timestamp display showing '00:00:00:000', and three unchecked checkboxes: 'OK to Overwrite', 'Enable COM1 trigger', and 'Post-Trigger Mode'.
- Footer:** A 'Load Last Capture' button.

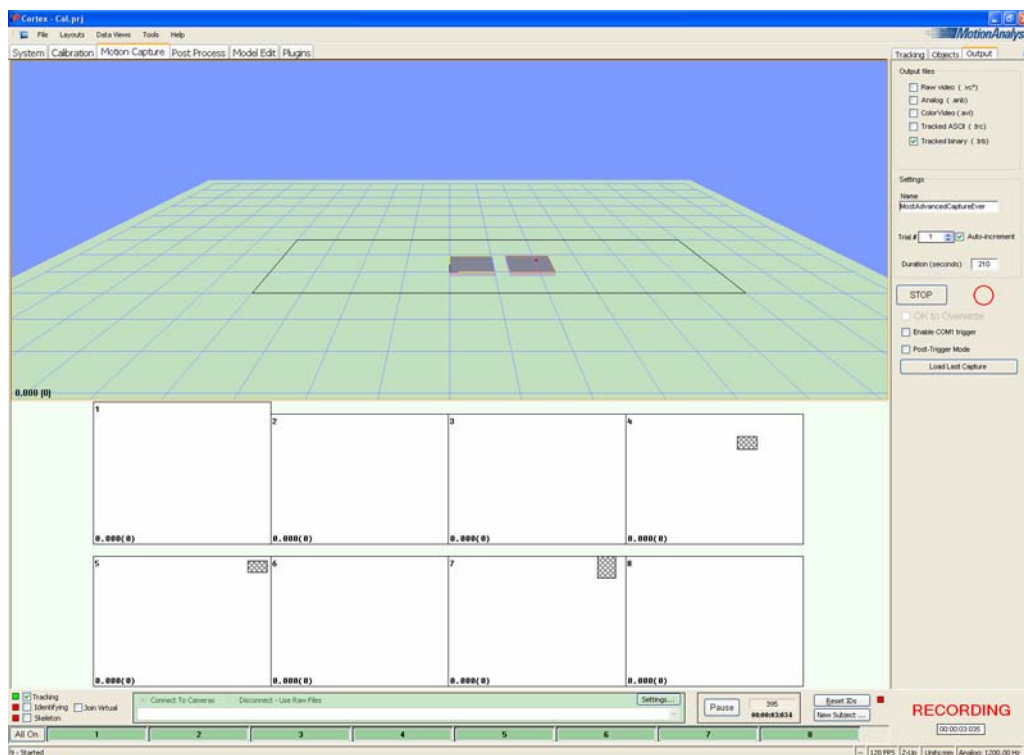
Output Files	The Output files are the files generated during a motion capture session. This section of the Output panel allows you to choose which files are to be produced and saved in the project file directory.
Settings	Settings allows you to provide the specifics for your generated output files. This includes the name of your files, the trial number, and the duration of the trial. The duration lets you set the length of the motion capture, so there is little wasted time in the session.
Note:	VC, TRB, TRC, ANB, AVI, and TC files are all associated in the project by file name. If you rename the output files, they may not be recognized and they will lose their association to the project.
OK to Overwrite	This allows you to redo an existing file, once it has been saved.
Enable COM1 Trigger	When using an external trigger mechanism, you will need to check this box for the software to recognize it. You can install an external trigger by plugging it directly into your COM1 port on your Host computer.
Post Trigger Mode	When the Post Trigger Mode check box is activated, it enables the software to record the data from the end of the session, backwards based on the capture duration (X) that has been set (i.e. from the end of the data capture to -X seconds). This is useful for captures where there is no defined starting point or event, but the ending is well organized and smooth, and you would like to capture only the final moments. Note, this only works for tracked data, not analog or color video files.
Load Last Capture	This button will load the last TRB file captured with the current project file. By default, it will load TRB files first. If they don't exist (only TRC files were captured), then it will then load a TRC file.

Recording Data

After setting up the template, always be sure to save the project file (**File > Save Project**). Use the following procedure to record a motion capture session.

1. Select the Output panel.
2. Select the output file types to be generated.
3. Enter a name for the file.
4. Enter a trial number. This is optional and will self increment if multiple trials of the same name are recorded.
5. Enter the estimated time length for the motion capture recording.
6. Select **Enable COM1 Trigger** if you are using an external trigger.
7. Select **Post Trigger Mode** if you only want the last portion of your motion capture session. This function allows you to select from the end of the session, working backwards to a specified time point.
8. Click **Record**. You should see the word **RECORDING** in large red letters on the lower right corner of the interface. See [Figure 9-6](#).

Figure 9-6. Recording Data



9. When recording is complete, click **Load Last Capture** to replay the tracked data that was last recorded. This will automatically send you to the Post Processing mode.

Tracking Strategies and Tips

Speeding Up Tracking in Cortex

Changing the Max Speed to 30 mm/frame and the Max Prediction Error to 10 mm greatly enhances tracking for most normal speed data sets. While this needs to be adjusted up for high-speed trials, keeping a 3:1 ratio works well. This can cut your CPU load by 50% or more.

Here is what is under the hood, see the **Motion Capture >Tracking** panel.

Max. Speed

If a marker has NO track history (i.e. a new marker just found on this frame), how big of a sphere do we draw around its current location to look for its continuation in the next frame. This should affect only startup tracks. It can affect performance if it is TOO BIG by making the software check more points than are necessary. If it is TOO SMALL, it will not create contiguous tracks; the tracks will have many holes and lots of unnamed markers.

Max. Predictor Error

Max. Predictor Error determines a sphere around the projected (extrapolated) path trajectory into the next frame. If the marker is not found in that projected sphere, it is assumed to have disappeared. This should affect only continuing tracks with a history, which is the bulk of what is being tracked. This can have a big effect on performance if it is TOO BIG. If it is TOO SMALL, tracks will be broken up into smaller path fragments and there will be an excessive amount of unnamed markers.

Bring up the Task Manager and monitor the CPU usage. You can see this with trials you have already collected by selecting the Raw Video Files you collected, click the **Run** button. Change the above Tracking Parameters, press the **Run** button again and see the difference. Check that the **System > Misc** panel is showing the **From Raw Video Files** and the Real Time speed option is selected.

Topic	Page
Editing Tracked Data	10-1
Viewing Your Data	10-2
Joining Gaps in Data	10-5
Unnamed Markers	10-11
Post Process Tool Strip	10-12
Join Virtual	10-24
Data Painting	10-28
Time Lines	10-29
Analysis Graphs	10-30
Post Processing Strategies and Tips	10-33

Editing Tracked Data

The **Cortex** Post Process mode allows you to play back and edit tracked data stored in TRB (binary) and TRC (ASCII) files. Markers can be identified if they have gone unnamed during the recording session. Gaps and aberrations in data can be filled or fixed by hand, frame by frame, or by employing mathematical functions across entire sections of a data set. Up to ten operations can be undone if you make a mistake, but it is recommended that you save your work frequently.

Typically, an editing session requires having both the 3D View and the XYZ Graphs open at the same time in two different Graphics Panes.

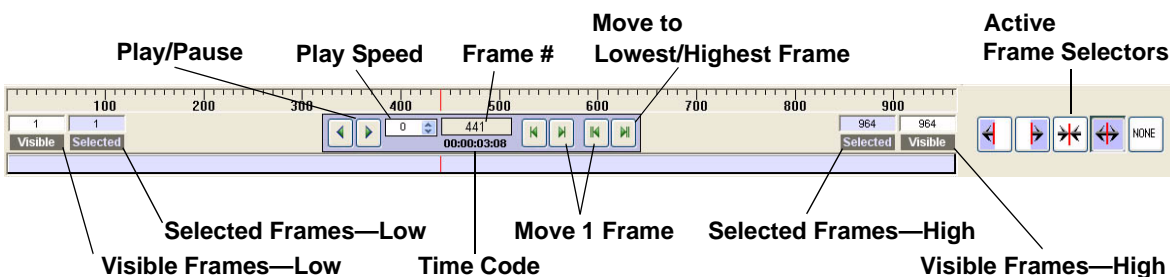
1. Click **Post Process** from the Mode Buttons.
2. From the Menu Bar, choose **Layouts > 2 Panes: Top/Bottom**.
3. Left-click in the Top Pane to select it.
4. Activate the 3D View by pressing **F3** on the keyboard or by choosing **Data Views > 3D View** from the Menu Bar.
5. Left-click on the bottom pane to select it.
6. Activate the XYZ Graphs by pressing **F4** on the keyboard or by choosing **Data Views > XYZ Graphs** from the Menu Bar. It will display X, Y, and Z tracked position data, and optionally, residuals, and the cameras that triangulated the markers.

Viewing Your Data

The XYZ Graphs will display none, any, or all markers you have selected in the Post Process tab. Selection of markers from a marker list is done with standard selection methods including **Shift** + click, **Ctrl** + click, and **Shift** + **Ctrl** + click. In addition, the top row of the marker list acts as a special selecting button. Markers can also be selected by simply clicking and **Ctrl** + clicking on markers in the 3D View. Click on the back button to reset the previous list of selected markers.

The Post Process tools heavily utilize the Post Process Dashboard controls. These controls are itemized and described as follows (from left to right in [Figure 10-1](#)).

Figure 10-1. Post Process Dashboard



Low and High Visible Frames

The Low and High Visible Frames define the lower and upper limits of the visible frame range. The Current Frame is never outside of these limits. Absolutely no identifying or editing can occur on frames outside of the visible range, with the exception of the join tools. Their values are found in the white Visible Boxes and can be changed by typing numbers into these boxes and pressing **Enter** on the keyboard.

Low and High Selected Frames

The Low and High Selected Frames are the lower and upper limits of the selected frame range. These values can be changed with a middle-click drag mouse drag action or by typing numbers into these boxes and pressing **Enter** on the keyboard. The Selected Frames region is shaded in light-blue. See [“Joining Gaps in Data” on page 10-5](#) for details. When editing data in the XYZ view, only data in the Selected Frames region will be affected. Areas outside of the Selected Frames are protected from the edit steps.

Play Forward Button/Pause

The Play Forward button (default hot key is the **>** key) plays forward through the data until the end and then repeat from the beginning. This also acts as a Stop button.

Play Backward Button/Pause

The Play Backward Button (default hot key is the **<** key) plays backward through the data until the beginning and then repeat from the end. This also acts as a Stop button.

Play Speed	Sets the playback speed of the tracked data, in ratio to the number in this box. a value of 10 will play the tracked data 10 times faster than normal speed. A value of -10 will set it at 10 times slower than normal speed. The limits are 100 and -100.
Time Code	This shows the time code (HH:MM:SS) for the tracked data. If a time code reader and a time code generator are used when data is collected, the time code at recording time is shown. If no time code reader or generator is used, the time code starts at zero for each trial.
Frame Number	The Frame Number is the frame that is currently seen in the 3D View and is marked with a full height red line on the XYZ Graphs. The Current Frame number is found in the very center of the Post Process Dashboard.
Move 1 Frame Button	The Move 1 Frame Button (default hot key for forward is the F key, reverse is the S key) moves the Current Frame forward or back by one frame. This also acts as a Stop button.
Move to Lowest/ Highest Frame	This sets the current frame to the lowest or highest visible frame.
Time Zoom Slider	The Time Zoom Slider sets/indicates the Low and High Visible Frames. Double-clicking on this control “unzooms time” or expands the Visible frames to encompass all of the frames in the data set. If you only want to work with a specific range of frames, right-click on this slider to lock/unlock visible frames.
Selecting Frames	<p>To the right of the Post Process Dashboard controls are five Frame Selector buttons. Refer to Figure 10-2.</p> <ol style="list-style-type: none">1. Select Backward selects from the Current Frame to the Low Visible Frame.2. Select Forward selects from the Current Frame to the High Visible Frame.3. Select Current Frame selects only the Current Frame.4. Select Visible Frames selects from the Low Visible Frame to the High Visible Frame.5. Select None Safe Mode—nothing is selected.

Figure 10-2. Selecting Frames Buttons



Clicking any one of these buttons sets the Frame Selector mode that you can return to at any time by pressing **Esc** on the keyboard. The Frame Selector mode is a User Preference. The default mode is Select Visible Frames. This will also highlight the selected area in blue.

Editing XYZ data is only done in the Selected Frames. The frames outside of the Selected Frames are protected.

Zoom In-Zoom Out

The XYZ Graphs right mouse pop-up menu includes the time zooming features (default hot keys are the **I** and **O** keys). These features zoom in time (frames) centering on the Current Frame. Zooming occurs more quickly by using the hot keys.

The best way to zoom into a particular set of frames is to select the frames in the XYZ Graphs by dragging with the middle mouse and then pressing the Zoom In Hot Key. Even finer control over zooming can be accomplished by pressing **Shift** + middle-clicking to independently set the Low and High Selected Frames and then pressing the Zoom In hot key (**I**).

The XYZ Graphs also allows you to translate the data vertically and horizontally. This is accomplished by holding the **Alt** key while clicking and dragging the cursor inside the X, Y, or Z display.

It is often helpful to zoom into the data's amplitude. Holding the **Alt** key and simultaneously pressing the left and middle mouse buttons zooms the data's amplitude. The marker that was closest to the pixel on the display where zooming began becomes the Target Marker. Its data is centered either to the data in the Current Frame or optionally to the data in the frame that the cursor was on when zooming began. Data for this marker will remain centered on the screen at all times unless you forcibly translate it off the screen using **Alt** + click and drag.

Unzoom is a means of resetting the display such that zoom and translate values are equal to zero. The default Hot Key is **U** on the keyboard and it is an XYZ Graphs right mouse pop-up menu item.

Target Marker

Picking a marker out of a crowd of data is done by double-clicking directly on a marker's data line. This action will deselect all other markers leaving only the display of the Target Marker.

Scaling

Other important view options that are general in nature are described here. These options can only be accessed as XYZ Graphs right mouse pop-up menu items.

- **Auto Scale** dynamically scales the display to accommodate data in the visible frame range.
- **Uniform Scale** the display such that X, Y, and Z conform to a uniform range.
- **Show Residuals and Cameras** shows residuals and cameras along with XYZ data.

Joining Gaps in Data

Join functions are not always confined to selected frames as a convenience. For example, if you select only one frame in a gap of marker data, tools intended to fill that gap will seek out appropriate endpoints to that gap, store all necessary data in an undo buffer, and effect a repair to that gap without requiring you to tediously hand select the appropriate endpoints.

The smooth function smooths data within the selected frames with a Butterworth Filter algorithm. This is a low pass (high block), two-pass, 4th order, zero phase shift filter. This data can be spikes—created by frames in which a marker has experienced an acceleration greater than or equal to a selected value, or gaps—missing data.

Manually selecting frames is done by dragging the mouse in the XYZ Graphs with the middle mouse button pressed. Low and High Selected Frames can be independently picked by pressing the **Shift** key and middle clicking on the XYZ Graphs.

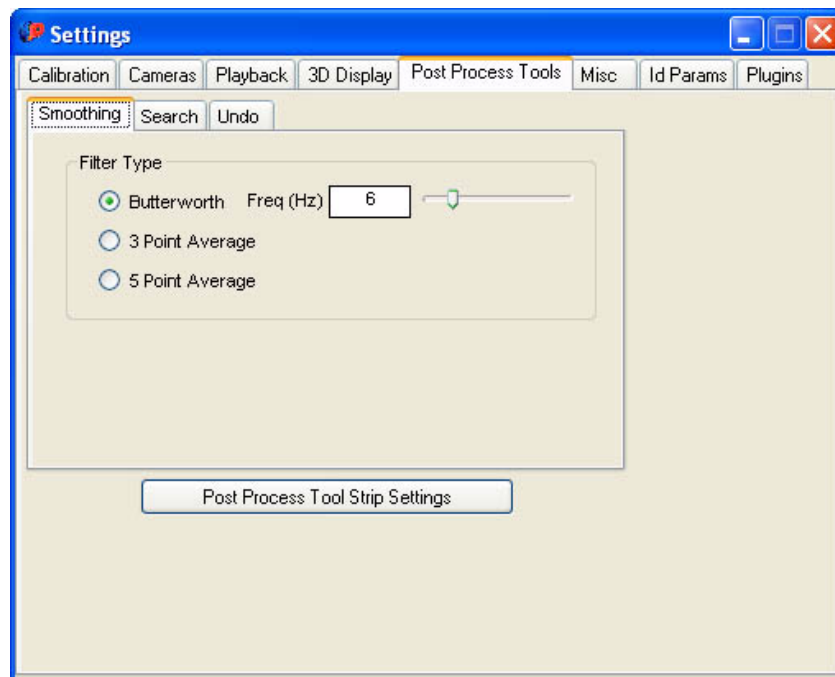
Select All Frames (default Hot Key is the **A** key.) displays and selects all frames in the data set. This is also a right mouse menu item on the XYZ Graphs.

Filters

The Eagle and Hawk digital cameras generate extremely clean, noise free data. For the majority of data captures, it is never necessary to modify the data by filtering or smoothing. Occasionally, however, it is useful to remove artifacts in the motion capture data. This can happen in the case of captures which contain a high number of marker occlusions or a large amount of marker merging (as it frequently happens with face tracking, for example).

For these purposes **Cortex** provides 3 different smoothing filters that can be applied to tracking data. Each filter affects only the currently selected markers over the currently selected sample range. All three dimensions (X, Y and Z) of each marker are smoothed. To access the options dialog, select the **Post Process > Options...** button.

Figure 10-3. Tools > Settings > Post Process Tools > Smoothing Tab



Butterworth Filter

The Butterworth filter is a low band-pass filter with excellent mathematical characteristics for biomechanical motion. The purpose is to remove high frequency motions (motions that are too fast for a person to actually perform) while leaving intact the frequencies of motion normal to human movement. The user has the choice of selecting how aggressively the filter will smooth the motion by choosing a frequency value. Lower values will cause a very smooth result while higher values will remain truer to the original data.

Figure 10-4. An Example of Original, Unfiltered Data with Some Unwanted Error

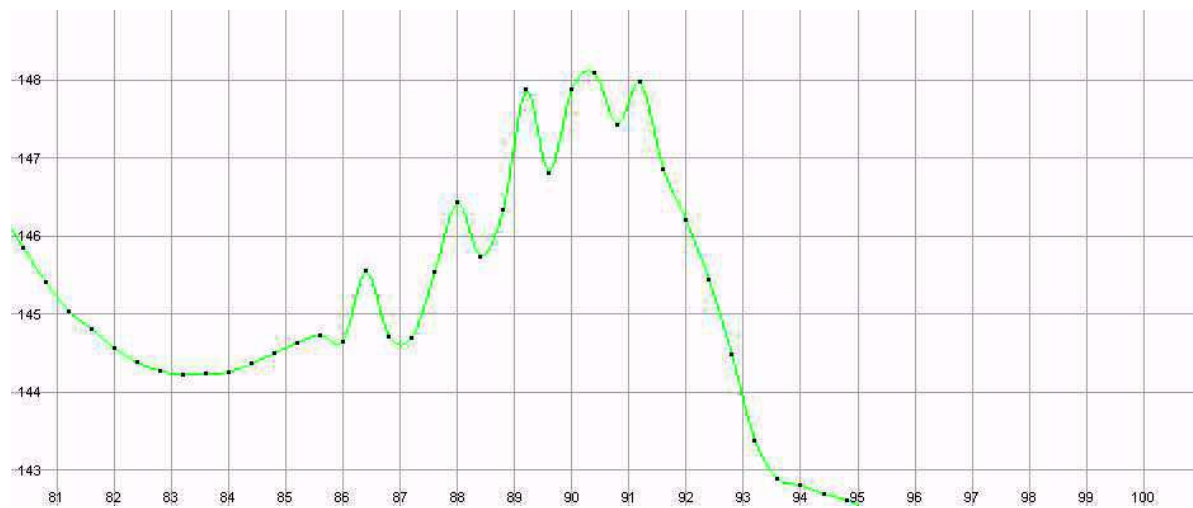


Figure 10-5. The Curve After an Application of the Butterworth Filter, with an Input Freq. of 3

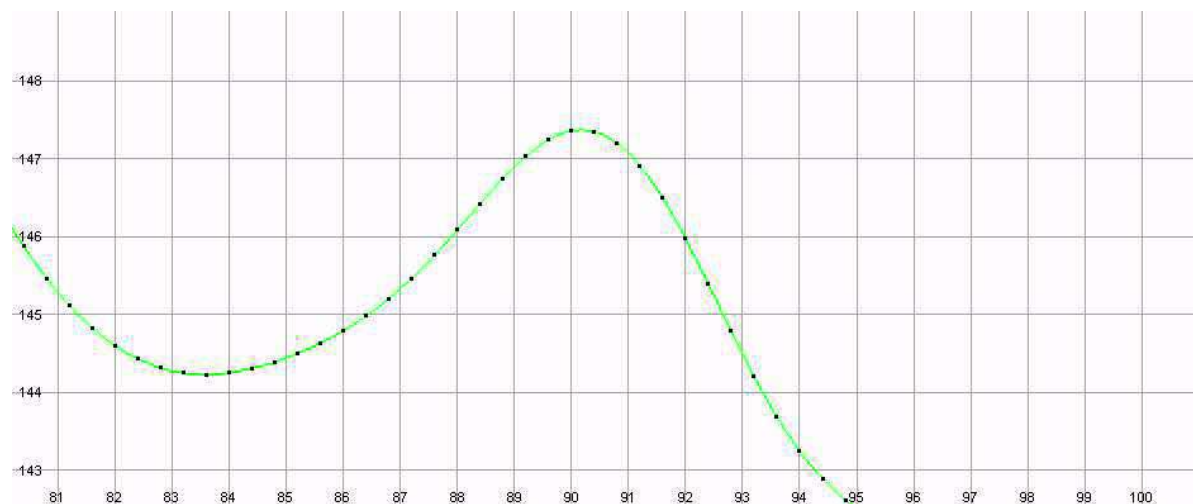


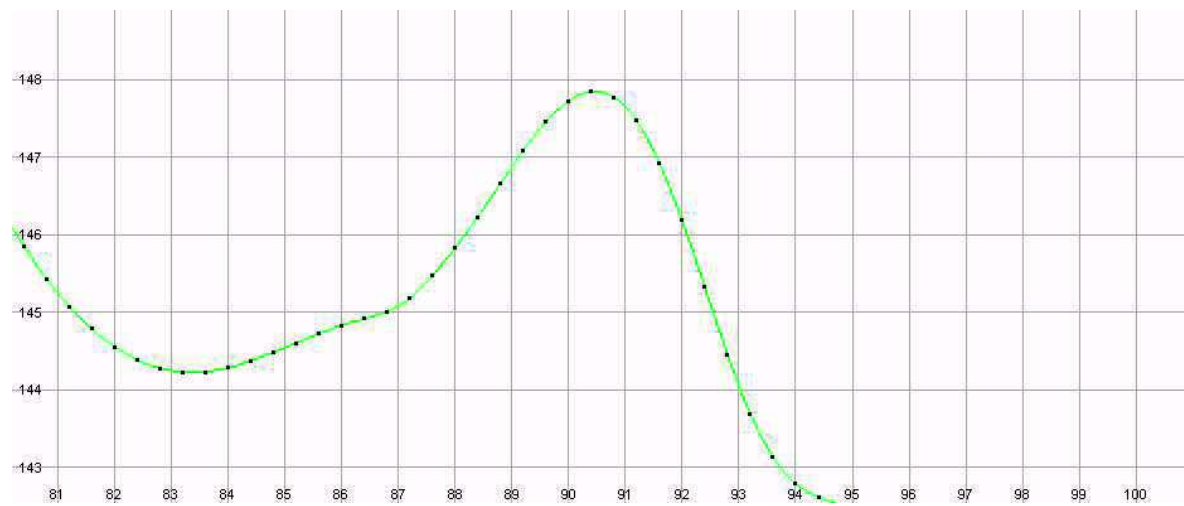
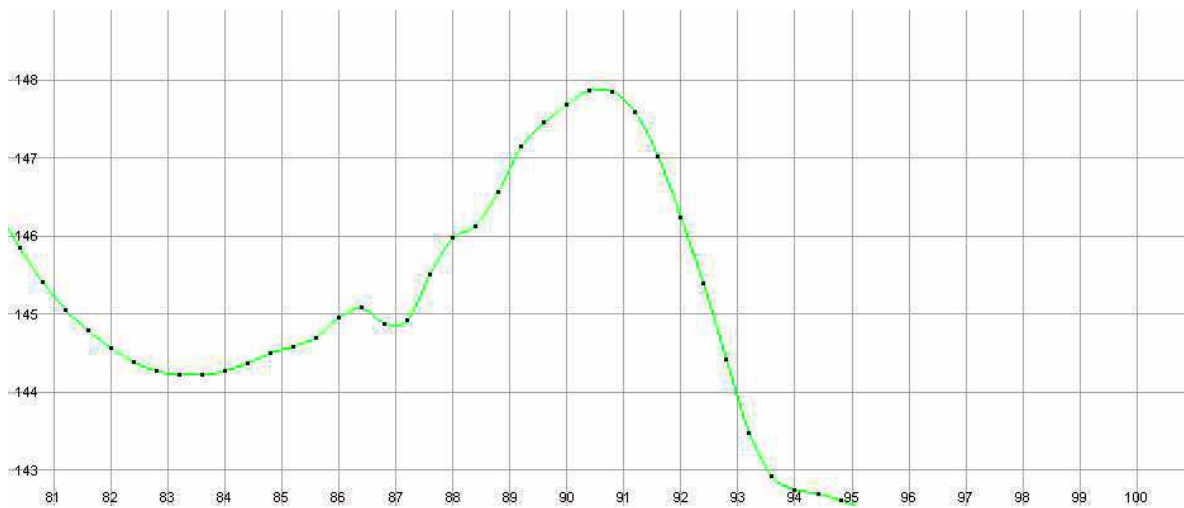
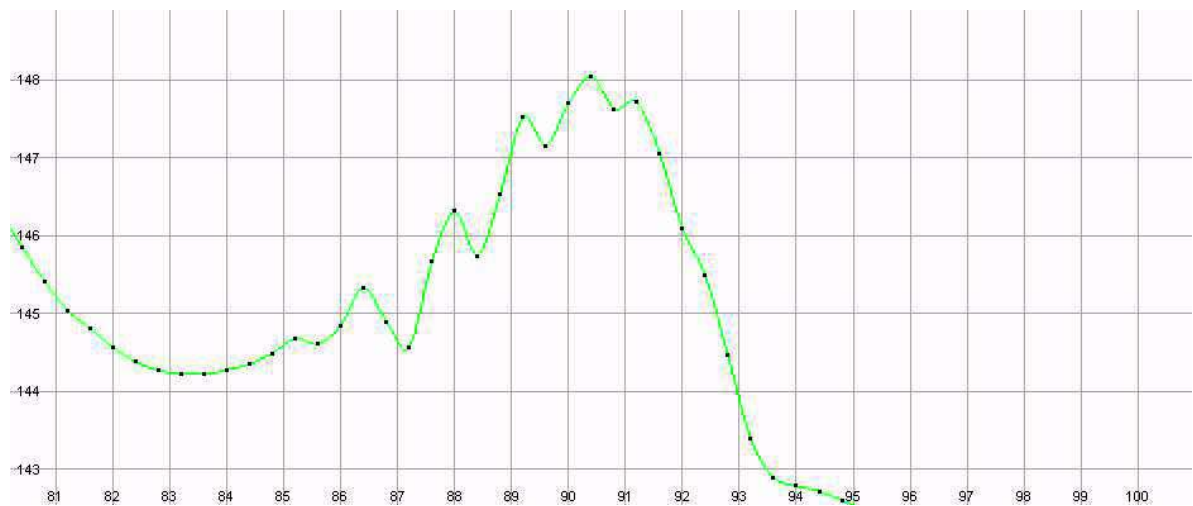
Figure 10-6. The Curve After an Application of the Butterworth Filter, with an Input Freq. of 6**Figure 10-7. The Curve After an Application of the Butterworth Filter, with an Input Freq. of 12**

Figure 10-8. The Curve After an Application of the Butterworth Filter, with an Input Freq. of 18

As the previous sequence of images shows, the input value to the Butterworth filter noticeably changes the result. The first example, a frequency input of 3, shows a lot of smoothing applied to the curve while the last example shows very little change to the curve. Depending on your needs, these might be appropriate levels of change. For most purposes, however, values between 6 and 12 work very well as can be seen in the middle two images. In these two examples, the noisy part of the data has been removed while the overall characteristics of motion have been retained.

Details of the Butterworth Implementation in Cortex:

The Butterworth filter does have that complexity of having to use "extra data" at the end points.

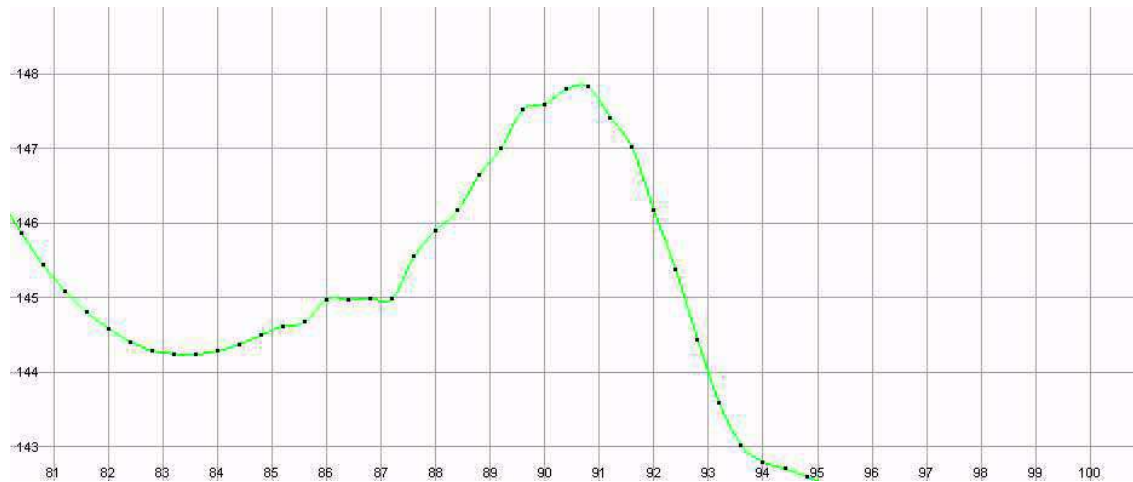
The steps are

1. Linearly transform the data to be filtered so that the first and last points are zero (0).
2. Double the number of points to be filtered. The data goes into the middle (2/4, 3/4).
3. Mirror image the first half of the data into the first quarter of the expanded points.
4. Mirror image the second half of the data into the fourth quarter of the expanded points.
5. Finally, run the two pass, fourth order filtering and then restore the data through the inverse transform done in step 1.

3-Point Average and 5-Point Average Filters

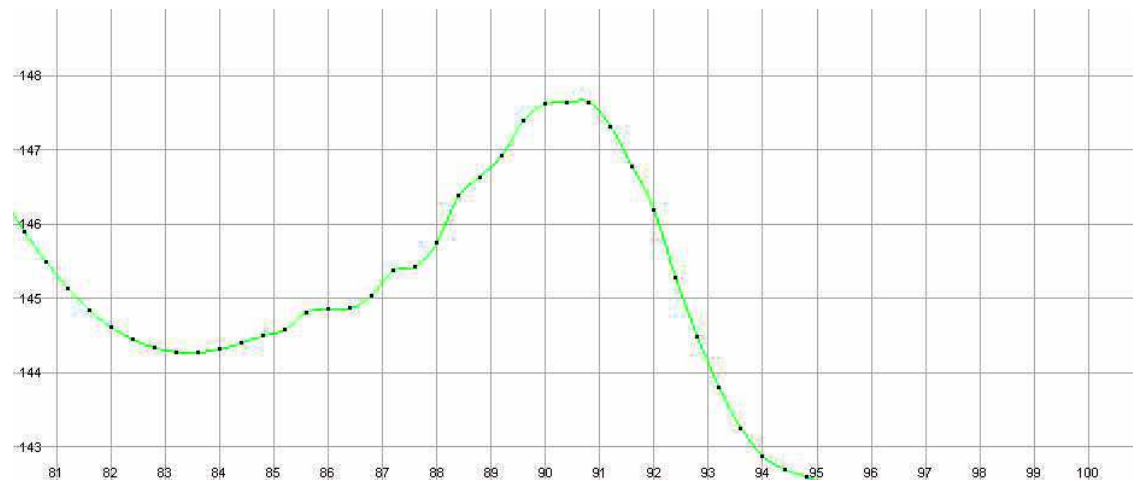
Two other filters, the 3 Point Average and the 5 Point Average filter are provided as an alternative to the Butterworth filter. In some circumstances (particularly with facial data), these can provide better results. The 3 Point Average smooths the data by taking a data point on either side of a given, original data point and averaging their values into the original one to create a new data value for that sample. This filter provides a moderate amount of smoothing as shown below:

Figure 10-9. The Curve After an Application of the 3 Point Average Filter



The 5 Point Average filter works just like the 3 Point Average filter except that it uses 2 data points on either side of the original data point to produce a new value. Since the width of the filter is wider the results are more aggressive which creates more smoothing, as seen here:

Figure 10-10. The Curve After an Application of the 5-Point Average Filter



The type of smoothing you choose depends on your needs and how much (and in what way) you want to change your data. It is perfectly reasonable to make successive applications of the filter(s) to affect the data in various ways. The number of possible combinations are extremely high so some experimentation will be necessary to find the right one for you.

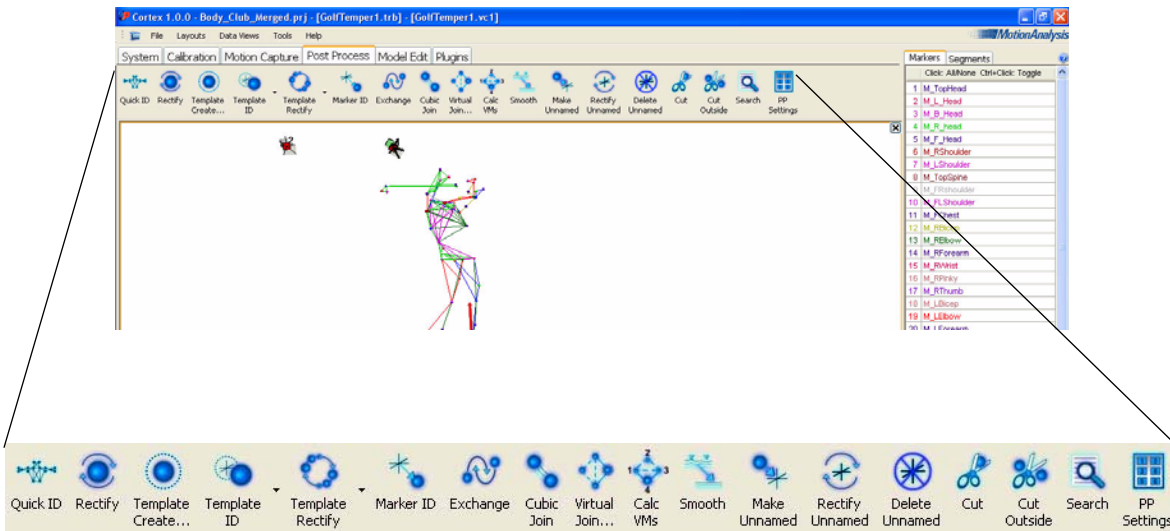
Unnamed Markers

Unnamed markers are defined as unidentified markers that are either real or not real. Some unnamed markers represent good data, yet were unidentified during a motion capture session. Others are called ghost markers and should be deleted. Ghost markers can also be removed from future data captures by going to the Tracking panel in Motion Capture, and setting the **Min. Cameras to Use** (minimum number of cameras to use) to **3**.

Note: Caution should be taken here as this process may also eliminate good data.

Post Process Tool Strip

Figure 10-11. Post Process Tool Strip



All of the Identifying tools are accessible using hot keys, panel buttons, and right mouse menu items on the 3D View and XYZ Graphs.

Note: You can rearrange, add, and delete the icons by right-clicking on the Post Process Tool Strip. To move the icon to a different location on the Tool Strip, press the **Alt + Left-Mouse** button and drag it to the new location. The Tool Strip order is remembered the next time **Cortex** is launched, allowing for user customizing and convenience. Also, any of these functions can be stored and accessed under Hot Keys.

Default Icons

The following are the default icons as set by the initial installation of the Cortex software.

Quick ID



Identifies the selected marker, identifying all markers one by one, according to the list. It will normally select with auto incrementation (auto increment).

The user identifies the markers in the MarkerSet list, one marker at a time from the top of the list.

The marker list will auto-increment after a marker is identified.

Rectify

Re-identifies missing markers (gaps) in a determined frame range. For more information on the Rectify functions, refer to [“Rectify Functions: What They Do and When To Use Them”](#) on page 10-19.

It is also used for cleaning up the Initial Pose for making a template when you have no template to start with. Takes ALL markers on the current frame (regardless of the All vs. Selected radial button), measures the linkages on the current frame and uses those measures to automatically sort markers into the correct marker slots.

Characteristics of Rectify:

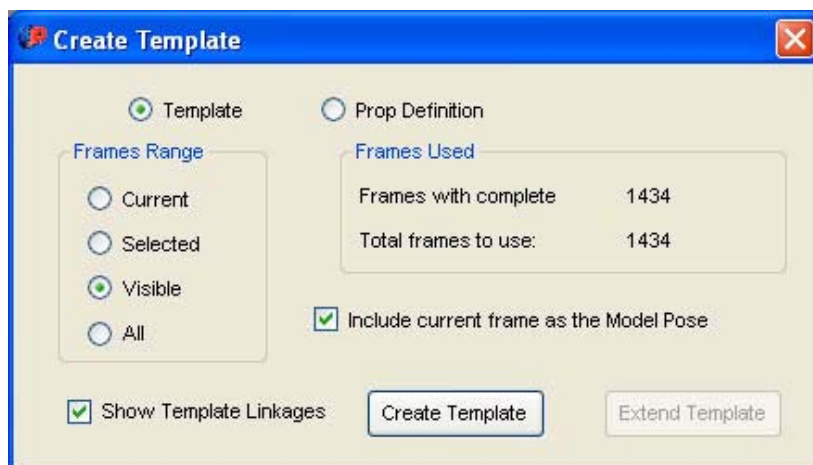
- Uses all markers, Named and Un-named
- Works only on the Highlighted XYZ Selected Time Range
- Uses the Named marker linkages and XYZ path continuity
- It will switch Named markers (Named markers are not automatically locked)
- Adjusts Linkage lengths dynamically to fit the data (including mistakes)
- Uses the **Tools > Settings > Identifying Parameters** function (typical)

Template Create...

A template for identification can be created which will be used to auto ID markers in Real Time, for Template ID, and Template Rectify.

For more information, refer to [“Building a Template”](#) on page 9-5.

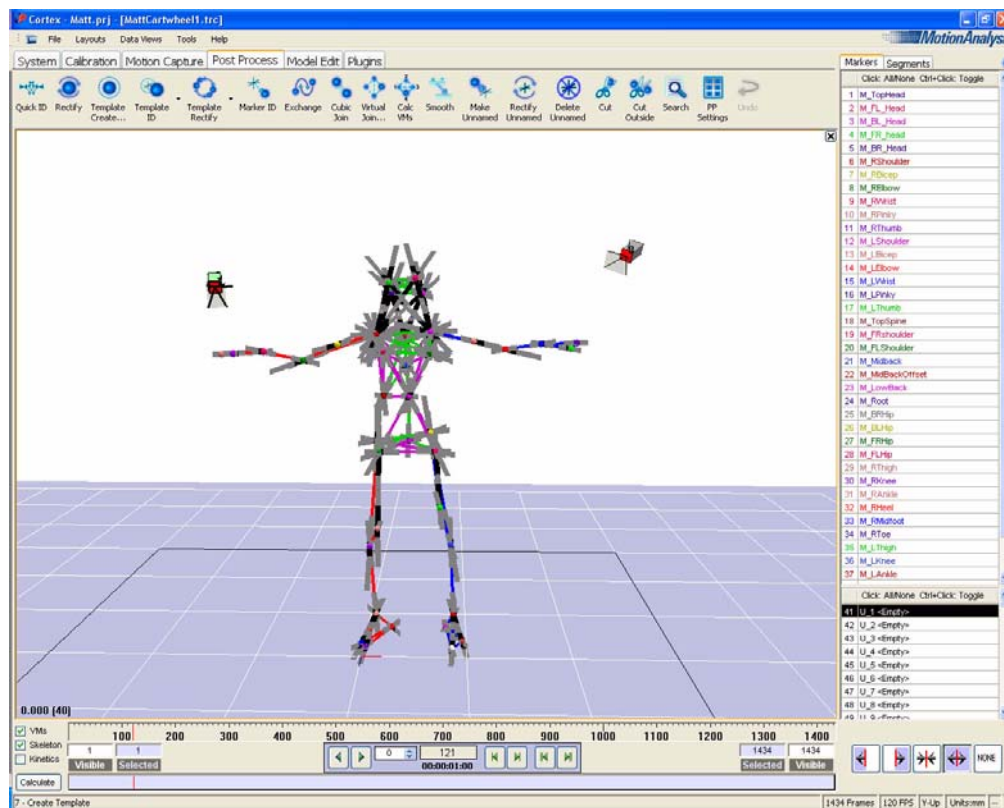
Figure 10-12. Create Template Interface

**Show Template Linkages**

This option selects all linkages and displays them in the following manner:

- Black parts represent measured linkage information for use with the Identifying Template for Template ID and Template Rectify. This is set or extended when you Create or Extend the Template.
- Grey Parts represent the “Extra Stretch” used by the Template ID and Template Rectify. This is set in the **Model Edit > TreeView-Links**. You can set individual links or groups of links.
- When you create a new link in the **Model Edit > Markers Panel**, the default values for the Extra Stretch is 15%, can be changed in **Model Edit > TreeView**.
- For compatibility with previous versions of **EVaRT** software, project files with undefined templates were always given extra stretch values of 0.30 (30%), which is larger than is now optimal. Reading older project files with undefined links will be given Extra Stretch value of 0.30 and can be changed by the user selecting those links, setting the Extra Stretch and saving the project file.

Figure 10-13. Show Template Linkages



This “feathered stick figure” can be used to look for possible problems in the template such as:

- Asymmetry—Unless the person is doing asymmetric activities, the template should be fairly symmetric in both the black and grey bands.
- Black lines that are too long—This can be caused from a mis-identified marker or a marker switch that got measured and stored in the template with the Create Template or Extend Template functions.

Template ID

Uses the template to ID all markers in the current frame.

**All Markers Pull Down Selection**

Allows you to select all markers at once for Identifying.

Selected Markers Pull Down Selection

Allows you to select specific markers for Identifying.

Template Rectify

Uses the defined template and identified current frame to ID markers over the selected frames.



Option: All or selected markers.

Marker ID

The user selects a marker in Marker Set list to identify that marker in the 3D view. Marker ID is the same as Quick ID, without the auto increment feature.

**Exchange**

Exchanges the 3D positions of two selected markers.



Exchange requires that exactly two markers are selected. The data is exchanged between the markers within the selected frames.

Cubic Join

Calculates the values to place in the gaps with a cubic spline. If you manually select the endpoints of the gap before executing the join, the function will fill the gap with a linear interpolation because the second derivative at the endpoints equals zero.

**Join Linear**

Selecting this will automatically fill a gap in selected frames of the XYZ marker data with linearly interpolated data.



Virtual Join

The assigned Virtual Join definitions will be applied to XYZ marker data for the selected markers over the selected frames. For all information regarding Join Virtual, refer to [See “Join Virtual” on page 10-24](#).

Calculate Virtual Markers

This calculates the locations for all assigned virtual markers over the selected frames. For more information, refer to [“Virtual Markers” on page 11-17](#).

Smooth

Smooths data within the set frames with the selected filter type. A Butterworth, 3-pt moving average, or 5-pt moving average filter can be applied to the selected markers over the selected frames. Filter selection is found in the **Tools > Settings > Post Process Tools > Smoothing** form. For more information, refer to [“Filters” on page 10-5](#).

Make Unnamed

The selected marker(s) will become unidentified over the selected frames.

When selected, the marker's data will be moved to the first available unidentified marker slot.

Rectify Unnamed

Makes all unnamed/unidentified markers into contiguous paths to follow through the capture sequence. For more information on the Rectify functions, refer to [“Rectify Functions: What They Do and When To Use Them” on page 10-19](#).

Delete Unnamed

Deletes all unnamed markers over all frames.

Cut

XYZ marker data within the selected frames, including endpoints, will be cut out and is available for Paste.

Cut Outside

Cuts the data outside of the selected frames exclusive of the endpoints.

Search

Finds gaps and/or spikes throughout the data set. The current frame will be set to the first gap or spike found in either the first selected marker on the marker list or all of the markers. See the **Tools > Settings > Post Process Tools > Smoothing** tab.

The Acceleration at Spikes function will indicate the frames in which a marker has experienced an acceleration greater than or equal to the selected value. The indicator appears as a carat (V) at the top of the XYZ Graphs.

PP Settings

The PP Settings button opens a form that lets you set the following:

1. Smoothing filter type
2. Search: gaps and/or spikes
3. Undo buffer

Undo

Undo retrieves data affected by the most recent Edit or ID function and places it back into the data set. **Cortex** supports ten levels of undo. This feature can be disabled or cleared on the Post Options form. If you get the message that an Undo function may not execute, you may need to clean your Undo buffer. This can be found in the **Tools > Settings > Post Process Tools > Undo** tab.

The Memory Gauge lets you know when your computer is running out of memory to store edits in the undo buffer.

Additional Icons**Hide Markers**

Hides selected marker(s) from the 3D view.

Hidden markers will have an **(H)** in front of the marker name in the Marker Set list.

Unhide Markers

Unhides the hidden selected 3D view markers.

**Rigid Body Rectify**

Uses the selected markers to ID unnamed markers through the capture sequence. For more information on the Rectify functions, refer to [“Rectify Functions: What They Do and When To Use Them”](#) on page 10-19.



Rigid Body Rectify and Template Rectify assume that all the current marker identifications are correct. They are intended for continuing the identification process without undoing previous work.

Rigid Body Rectify is a tool that could be considered a "stand-alone" tool. It does not use anything from the marker set definition at all. When the tool is activated:

1. The three, or more, selected markers are dynamically turned into a "Rigid Body" definition and measured
2. The previous frame and the current frame are then used to predict the next frame
3. Identify the frame

This stops when less than three markers of the original selected markers is identified.

Note: If one or more markers are already correctly identified, then that can help prevent errors.

This has been used to identify the entire body.

1. Select ALL the markers (minus the obscured ones). The starting frame must be identified manually.
2. Press **Rigid Body Rectify**
3. Go forward to the frame where the misidentification occurred
4. Make unnamed
5. Repeat steps 2 through 5.

Copy

Data within the selected frames, including endpoints, will be copied and available for Paste.



Paste

Cut or copied XYZ marker data will be inserted into the data set beginning with the current frame.

Rigid Body Join

The rigid body join feature has been created for rigid objects with 4 or more markers per segment. For rigid or semi-rigid objects such as swords, spears, head markers, torso markers, multiple markers on a basketball, it is convenient to use this feature to join across missing marker data. You must select a starting frame where all markers that you select are all present and part of a rigid body. You then select a range of frames on which you wish this to operate. Select **Rigid Body Join** and it automatically joins across the missing marker data.

**Rectify
Functions: What
They Do and
When To Use
Them**

There are three main Rectify tools for naming markers and propagating the names through time (Rectify, Rectify Unnamed, and Rigid Body Rectify). They are not for generating XYZ data from the 2D camera views, which we call Tracking, but they are very useful for Identifying the tracks or markers. Rectify means to “make right” or “set right”. All of the Rectify functions start at the Current frame and go forward in time first, then backwards from the current frame. For more information, reference these functions in the [Post Process Tool Strip](#) section starting on [page 10-12](#).

*Template Rectify*

In Post Processing, use this when you have a reasonably good template. Work from known good frames into unknown and difficult frame ranges. Uses the template from the Create Template item. A template can be of one or more frames, should represent characteristic motions to be seen between markers, and is a measure of the min. and max. linkage lengths for Named markers.

The characteristics of Template Rectify are as follows:

- Uses only the Template information to move markers from the Unnamed slots to the Named slots
- Uses All Markers or Selected Markers according to the settings.
- Works only on the highlighted XYZ selected time range, starting on the current frame going forward, then backward from the current frame
- Protects all named markers, will not switch them
- Works only to move Un-named markers into the Named marker slots—here, all markers are locked
- Does not use Linkage Stretch Parameters in **Motion Capture > Tracking**

Rigid Body Rectify

In Post Processing, it is used when you have bodies crashing into each other where linkage lengths can get very distorted and Template Rectify

can give results which may require some editing. Rigid Body Rectify is for very tough trials where you can have the software look for a rigid body with markers on it and it tends not to make mistakes and hence does not require a lot of editing after the fact. Work from a frame with known marker IDs to difficult areas. A Rigid Body can be any set of 3 or more Selected Markers. The software measures the markers with respect to each other and looks for this pattern in the Un-named marker to automatically assign them names. It will stop if either of two conditions are met:

1. It cannot find at least 3 of the markers of the Rigid body on a frame (so use more than 3 marker if you can)
2. The measurements stretch too much

It can be re-started on a new frame if needed. The selected markers must be identified on the Starting Frame.

The characteristics of Rigid Body Rectify are as follows:

- Measures all markers in the selected Rigid Body on the Starting Frame
- Uses All or Selected Markers (mostly used for Selected Markers)
- Protects all named markers, will not switch them.
- Works only to move Un-named markers into the Named marker slots—here, all the markers are locked
- Does not use Linkage Stretch Parameters in **Motion Capture > Tracking**

Rectify Unnamed



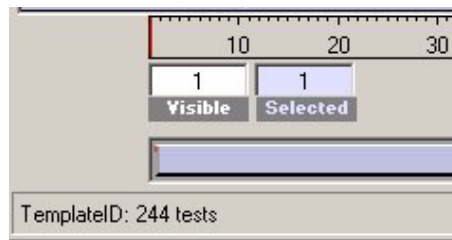
Rectify Unnamed sorts the Un-named slots into continuous paths based on path continuity, similar to the tracking function. No templates or linkages are used. Path segments separated by 10 or more frames are considered to be separate paths and will not necessarily be continuous. This is used to clean up the Unnamed path segments and can make the Marker ID function work more smoothly. It means you may not need the Rectify Always On check box in Marker ID and Quick ID Items. First try the Rectify Unnamed function, then try the Marker ID function for the problem areas.

Template ID



Uses the currently defined Template from Create Template to fit the linkages into the current frame's marker cloud. If it succeeds, it tells you the number of tests on the Status Bar (lower left side of the screen) it took to complete. It may also fail or time-out, in which case you should make sure all of the markers are present on the current frame or perhaps re-make your template. It only changes data on the current frame if it is successful. You can then use Template Rectify to get the correct IDs to other frames.

Figure 10-14. Template ID Details



How good is the Template? You can tell how good your template is by how many tests it takes for the Template ID function to work. A low number means it is working quickly (low is say 500). A high number (5,000 or more) means the software has to try very hard to ID the current frame. It stops trying after about 50,000 tries in Connect (Live) mode and “times out” and gives the message `Template ID: Timed out`. In Post Processing mode, it will not time out until 500,000 tries. That means it did not get to try out all possible linkages. A large number indicates a potential problem. It might be because the current frame is stretched beyond the template or in a much different actor position, or it might be because the Template is not very good. Try to use the Template ID on different frames and see if the number changes a lot. If you consistently get large numbers for the Template ID feature, try adding more links to your marker setup in the Model Edit tab.

Make triangles, especially triangles that will be fairly rigid during the movements. Lots of rigid triangles in your linkages make for solid Templates that have fast Template ID numbers. Triangles with equal sides can cause mis-IDs whereas triangles with unequal sides work much better. This will help to determine where to place markers on a person.

You must have all of the markers present on the current frame or you will get the status bar message `Template ID failed`. That means it did try out all possible linkages and could not get a match. Possibly the markers have moved or you need a better template.

Summary of When To Use What

Rectify

First, collect the trial from which you will make your template. It should be a simple trial with representative motions and not require any editing. The goal is to represent the min. and max. of each linkage in your model. Use Quick ID, then Rectify. If editing is required, consider taking another trial. Make sure there are no marker switches using the 3D and XYZ view, and then create a template using all of the (good) frames. Rectify may generate marker switches and you do not want those remembered in the template. It will haunt you later.

Template Rectify

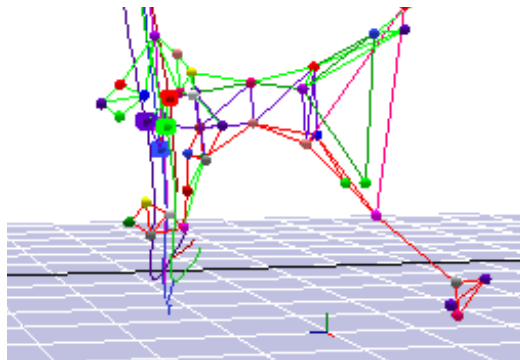
For most motion trials, the use of Template Rectify will do most of the work in correctly identifying the marker tracks. Template Rectify is preferred method to use to rectify since it protects the named marker tracks: they are locked. It also keeps the template rigid and seems less likely to

make mistakes. Correcting mistakes can take a lot of time. If there are some incorrectly ID-ed markers, it may be best to make all the markers Unnamed for all except the starting frame (which can be frame 1 or any other frame). To make all unnamed except frame 1, go to frame 2, select **All Markers**, then select the time range: Select **Forward**, and press **Make Unnamed** under the Identifying tab. Go back to frame 1 and press **Template Rectify**. On some complex trials where there is a crash or bang between people and or props, have the actors start in the T-pose, do the actions and return to the T-pose. You can then work the data from the start to the middle and also from the end to the middle. This working the data back and forth can save a lot of your time and not require that you hand ID many frames.

More on Templates and Template ID

After a big crash of people with markers or extreme movement, the markers may have moved and the template may not be as good. You can Extend the template by ID-ing after the crash, extend it based on one or more frames that have no mistakes, then try Template ID to see if the template holds on the new frames. Template ID tells you how good the template is working and it is the same template the Template Rectify uses. Also, think that Template Rectify gets used as much to un-identify tracks as well as identify them. If any link stretches beyond the allowable range (plus a small amount of give), then Template Rectify will cause the offending marker or markers to become unnamed, while it seems perfectly obvious to you what is right. If Template Rectify does this, manually ID that marker on that frame and Extend the Template (an option under the Create Template button). You can see this when the subject bends over and causes stretches that might not have been recorded when creating the template.

Figure 10-15. Template ID and Rectify Used for Link Stretch



Rigid Body Rectify

This takes the most time to use in that you must select a few markers and process a few frames, then repeat the process over the possibly many segments and frame sequences. If it stops, it means that the rigid body disappeared. You can re-start it again and it starts with new measurements on

the starting frame. Where it is very useful is where it can slug through some tough sequences. Start with something simple, like the 3 or 4 or 5 markers on the head. As with Template Rectify, it may be best to Make Unnamed all markers on all frames if there are any mistakes in the ID-ed data. If one of the head markers is incorrectly ID-ed as a neck marker, then Rigid Body Rectify will not see that as a candidate since it only looks in the Unnamed markers list. Process the data from both ends towards the difficult part, assuming there is a “crash” in the middle and clean data on both ends. After doing Rigid Body Rectify from the starting frames to the middle, then from the ending frames to the middle, use Template Rectify to go it again. For difficult data trials, this 1-2-3-4 combination will get you a lot of named markers for very little work.

Correctly Identifying Markers Automatically: Post Processing Mode

In the **Post Process** tab, the data is identified (or re-identified) by pressing the **Template Identify** button. This affects the current frame only and it is successful if:

1. The template is good and the data fits.
2. All markers are present.

To continue the correct identification to successive frames, you need to have the Max. Speed and Max. Prediction Error settings correctly set for your data.

Max Speed (mm/ frame)

This applies to when a marker first appears and is identified with the current template. To keep the correct identity into the next frame, the software checks to see if it has moved too much to be the same marker. It can move in any direction. The Max Speed parameter tells how much movement is allowed. It is measured in mm from the first frame, hence the units of mm/frame. If no marker is seen within this search sphere, then the target identify is not continued into the next frame. If a marker is found within the Max. Speed sphere, the target identity is continued into the second frame. If the number is set too small, tracking will slow down as the software tries unsuccessfully to find continuations of markers. This affects the first to second frame tracking time especially. If the number is set too big, you will see markers switch identities.

Max. Prediction Error (mm)

After a marker has a history of 2 or more frames of continuous identity in time, a track history allows the software to predict where the marker should be, based on a 2nd degree polynomial prediction. The software looks in a search radius of the Max Predictor Error about the predicted location for a continuation of each marker being tracked. The Max Prediction Error is usually set to about one-half of the Max. Speed parameter.

Join Virtual



Join Virtual is an extremely powerful editing tool used to fill gaps in marker data with simulated data based on the relationship (positional interpolation) with other markers on or near the particular problem segment. This accurately simulated information is a result of making four passes over the data in both Real Time and Post-Processing modes.

To use the Join Virtual function:

1. Find a gap within the position data of a marker. It is easiest to use the Search function (right click in XYZ Graphs and select Search).
If you are not using the Search function, select the gap area in the XYZ Graphs (middle-click and drag) for the problem marker.
2. Select **Join Virtual** in the **Post Process** tab. Verify that the Marker to Join is set for the marker you want to edit.

Figure 10-16. Join Virtual (Virtual Marker) Definitions



3. Click on **Origin Marker**. It will be highlighted in blue and will allow you to choose which marker to use as the marker that is most rigidly attached to the marker to join. See [“Origin Marker” on page 10-25](#).

Note: Select markers using the 3D View or the marker grids.

4. The function will then automatically jump to the Long Axis Marker input box. Continue selecting the proper markers for each remaining input box. Make sure they are all different. You cannot have two of the same markers in the Virtual Marker Join definitions.
5. Once you have defined the three definition markers for you Virtual Marker, click on **Join Virtual**.

Note: If data is missing for any definition marker (Origin Marker, Long Axis Marker, Plane Marker) in the frame field, the gap in data will not be completely filled. You will need to select a different definition marker that has data for that frame field.

6. Repeat [step 1](#) through [step 5](#) for all problem markers in your data set. You may also setup the Virtual Marker definitions for as many markers as you feel will be needed, prior to capture.

7. Selecting File > Save Project will save all Virtual Marker definitions you have set into the project file.

Real Time Streaming with Join Virtual Fill

If you will be continuing to capture motion using the same template, the Virtual Marker Join definitions are now resident with the project files and template. The Join Virtual check box can now be activated (on the Real Time Dashboard) allowing for Virtual Markers to be created in Real Time where data is missing for the markers you have set definitions. Thus, streamlining the editing process or post processing tasks.

Figure 10-17. Join Virtual Check Box



The concept behind the Join Virtual and the Virtual Marker definitions are the same and are much more stable and more useful than the classic Rigid Body data filling mechanisms. The reason is that you get to choose three markers in decreasing importance that determine the replacement data. These three markers are:

1. the Origin Marker
2. the Long Axis Marker
3. the Plane Marker

Origin Marker

The Origin Marker should be the marker that is most rigidly attached to the marker to join. If there are two choices, pick the one that is more stable on the bone segment. For example, the elbow marker is a good Origin Marker. It is usually attached close to a bone. The shoulder is also good for the upper arm segment, but not as good for the upper torso if the subject raises their arms.

For segments where you have multiple markers on a rigid segment, such as the head, it does not matter which marker is which. For example, if you have four markers on the head, each of the four can be defined by any order of the other three markers. But if you have only three markers on the head, the Top_Neck marker may well be used as the Plane marker for the Join Virtual definitions.

Long Axis Marker

The Long Axis Marker defines a straight line from the Origin Marker and the Join Virtual is not sensitive to changes in the length of this line.

Long Axis Marker Example

For the left Biceps, choose the L_Elbow as the Origin, the L_Shoulder as the Long Axis Marker, and the L_Wrist as the Plane Marker.

Plane Marker

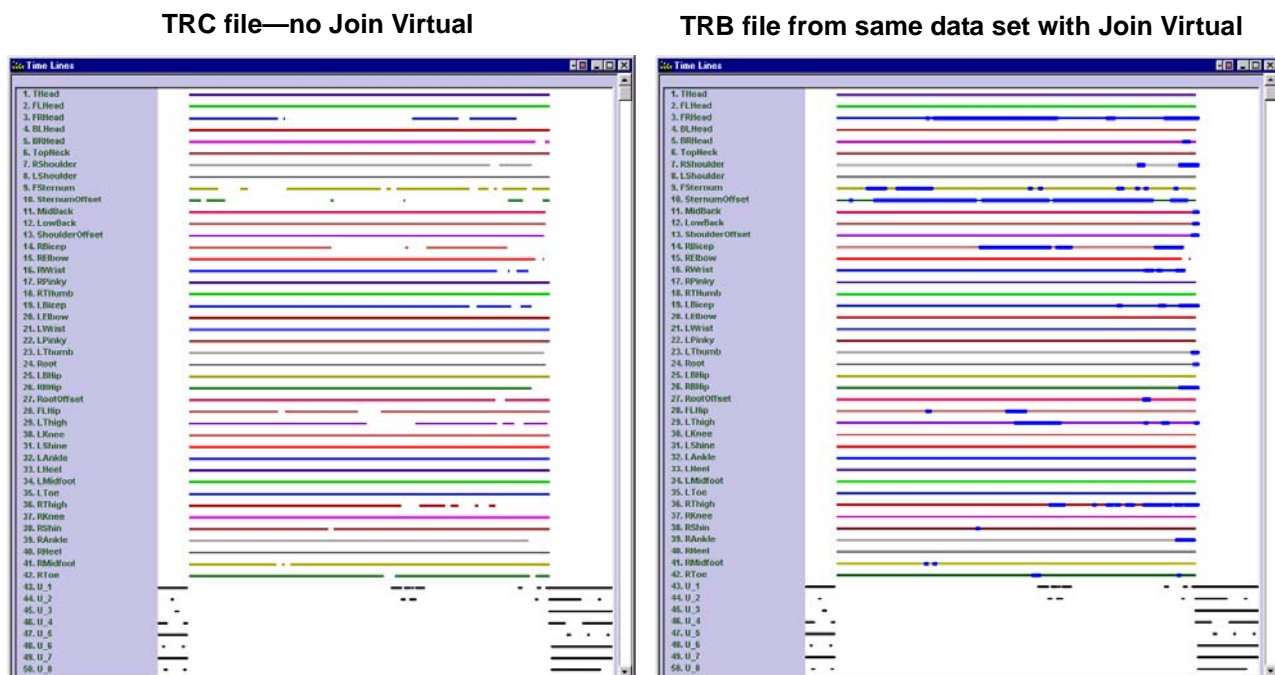
The Plane Marker is the least strongly coupled marker to the problem marker to Join. It defines only the rotation of the coordinate system located at the origin. Join Virtual and Virtual Marker calculations use the 3D offsets from the problem marker to the coordinate system's Origin Marker and apply that throughout the Join Virtual.

The results are often astoundingly good and can be used directly to speed up your animation pipelines. More study is recommended before applying these results to Biomechanics research and medical applications.

In short:

- The **Problem Marker** is the marker with the gap to be filled.
- The **Origin Marker** maintains a fixed distance to the problem marker.
- The **Long Axis Marker** defines a line to the origin marker.
- The **Plane Marker** defines a plane with the Origin and Long Axis Markers.

Figure 10-18. TRC File vs. TRB File With Join Virtual



To test the operation of this feature, define a Virtual Marker Join for the RHip marker based on 3 others that will remain visible. Cover the RHip marker (for example) and see if it appears in the 3D view. This can be done with a live person very easily, but if you do not have a setup available, go to the 2D view of a trial, mask out enough regions and cameras so that the RHip is no longer visible. It should appear in the 3D view if you turn on the Join Virtual feature.

Both the streaming and the post-process Join Virtual use a two-pass process to virtually join data across gaps. The data passes through the Virtual Marker Join function twice, with the second pass using filled or partially filled gaps that were not filled the first time.

Join Virtual Guidelines

Note: These guidelines are intended for an audience with a good knowledge of motion capture theory and practice. These are generalized guidelines only. Individuals may find that different definitions may work better for their particular applications.

For best results, it is recommended that you have at least three markers per effected segment. Ideally, for markers that have the possibility to become obscured (i.e. being covered up or lost between the ground and the subject's body) you will need to place markers on the opposite side of the appendage or body. For example, if a subject is laying prone on the floor, the back markers become obscured. If you anticipate this, you can apply more markers to the chest or front torso area.

For defining virtual markers, when possible, define and use markers that are always seen on that segment or neighboring segments.

If any data is missing from other markers in that segment, the original data will improve, but only if the dependent markers are present.

Head

A subject's head usually will have four or five markers. Missing marker data for the head is joined using Join Virtual definitions with any three of the other markers.

Upper Arm or Upper Leg

Typically the upper arm segment is defined by three markers: Elbow, Biceps, and Shoulder. Ideally, a fourth marker on the Triceps would be present. If not, a marker on the forearm can be used as the plane marker for Join Virtual data.

If the shoulder data is missing, you may use the markers for the top of the neck, sternum, or mid-back to calculate the Virtual Marker data for the shoulder.

Hand or Foot

If you know of movements that are going to obscure the markers on a hand or foot, you may want to set redundant markers on that particular segment.

For example, if all lateral side markers on a foot are obscured from a subject laying down in a prone position, you may apply redundant markers to the medial side to provide the data for that segment.

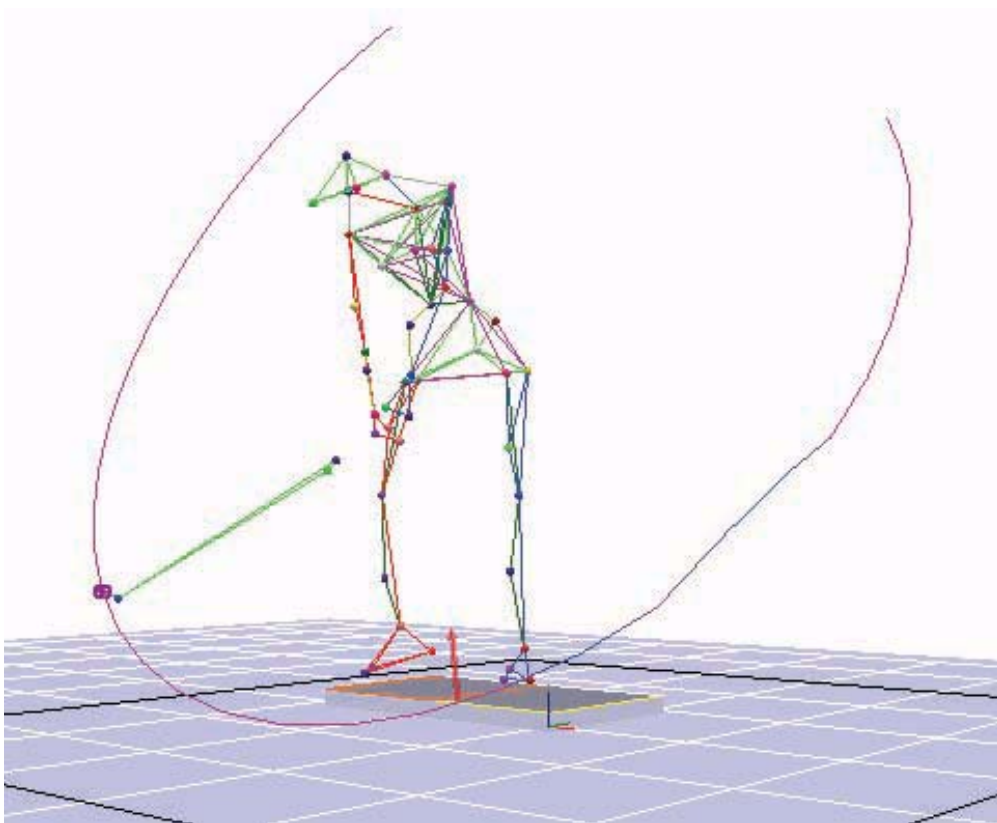
The hand will use the same technique, with maybe a few less redundant markers on the opposite side.

For an example of a project with Join Virtual definitions for all markers, see the **6Eagle_VirtualJoin** directory in **C:\Program Files\Motion Analysis\Cortex50\Samples**.

Data Painting

Data Painting allows you to directly manipulate the data in the XYZ Graphs pane. Simply press **Ctrl + Shift** and left-click to modify or add data directly on the screen. Since the outcome of using this feature is uncertain for each user, it should be used with care.

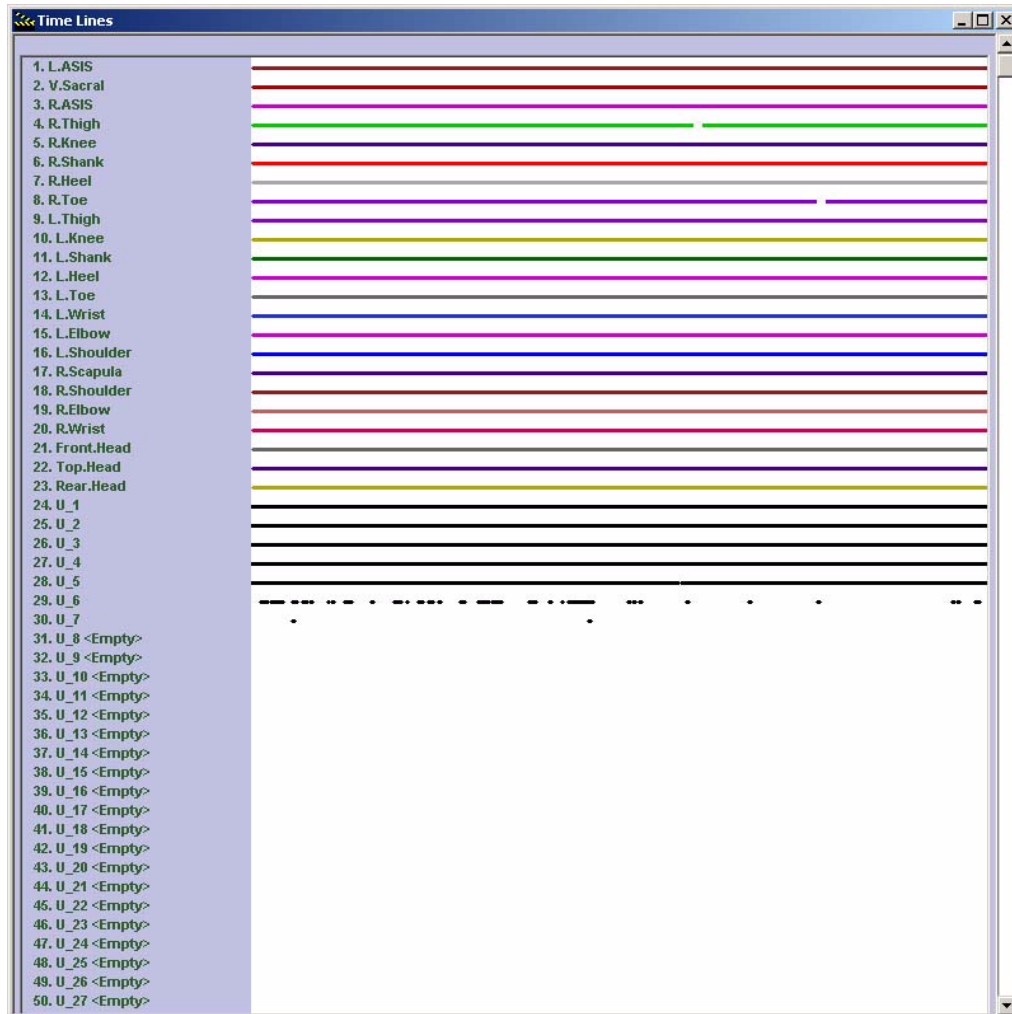
Figure 10-19. Data Painting



Time Lines

In the Menu Bar, select **Tools > Time Lines**. Time Lines provide a general overview of the quality of the data in the marker slots, showing any breaks in the stream of data for all markers.

Figure 10-20. Time Lines



Analysis Graphs

The Analysis graphs and their related control panel provide tools to analyze your realtime and post process data. This window has three tabs which calculate data for the following:

- Position, velocity, and acceleration
- Distance between markers
- Included angles

The Analysis graphs are activated by pressing the **F7** key.

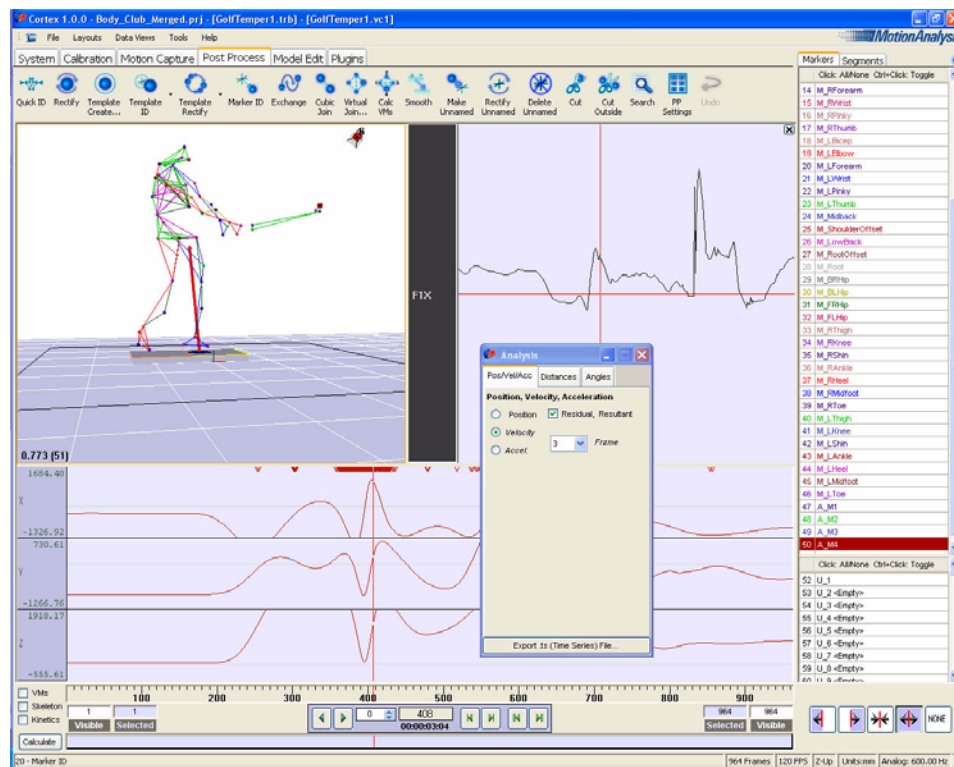
Position, Velocity, and Acceleration Tab

This tab creates graphs of the position data, calculated velocity data, or calculated acceleration data for up to 10 selected markers.

Note: Any number of marker data can be exported.

The number of frames used to calculate the velocity and acceleration data is set by the user. The number of frames used can be either 3, 5, 7, or 9. Using the higher number of frames to calculate the data will result in smoother output through noise reduction.

Figure 10-21. Position, Velocity, and Acceleration Tab

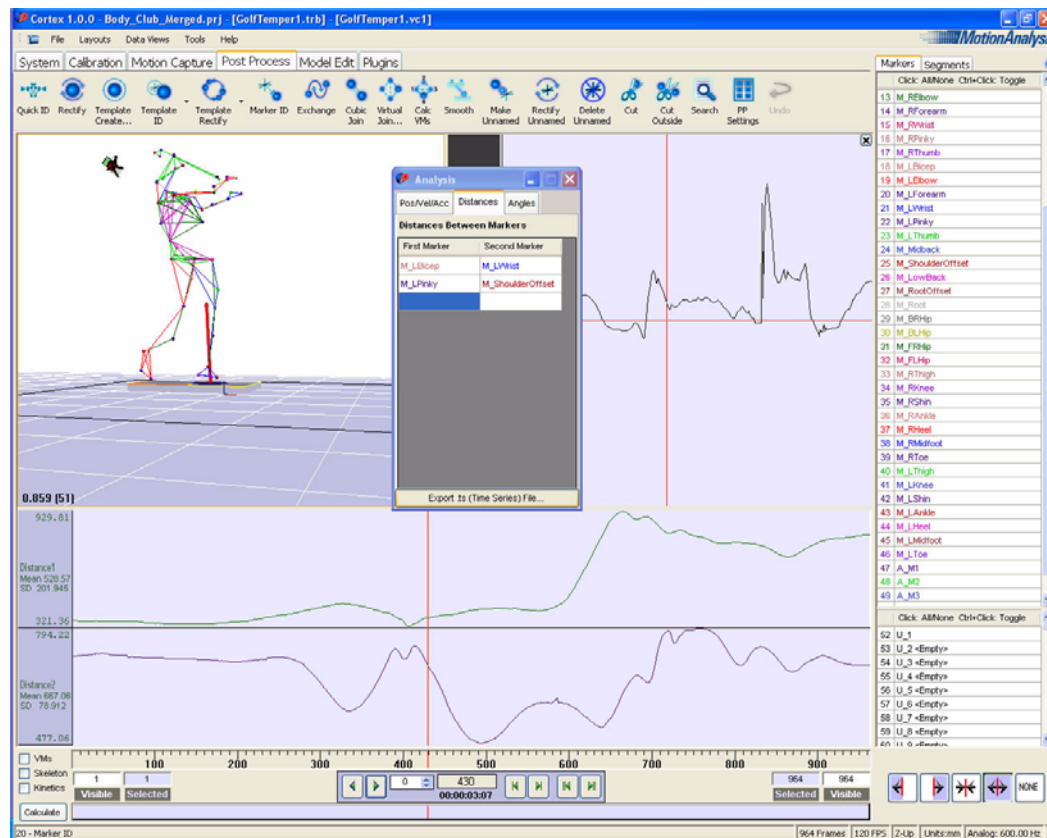


Distance Between Two Markers Tab

The Distance Between Markers tab shows the distance between two selected markers for each frame throughout the tracked data. You may select any two markers in the tracked data to be analyzed by clicking those markers in the 3D View or on the marker list grids.

To delete a pair of markers from the grid, click on the row and press the **Delete** key. You may select and delete several rows at once by pressing **shift + click** on the rows and pressing **Delete**.

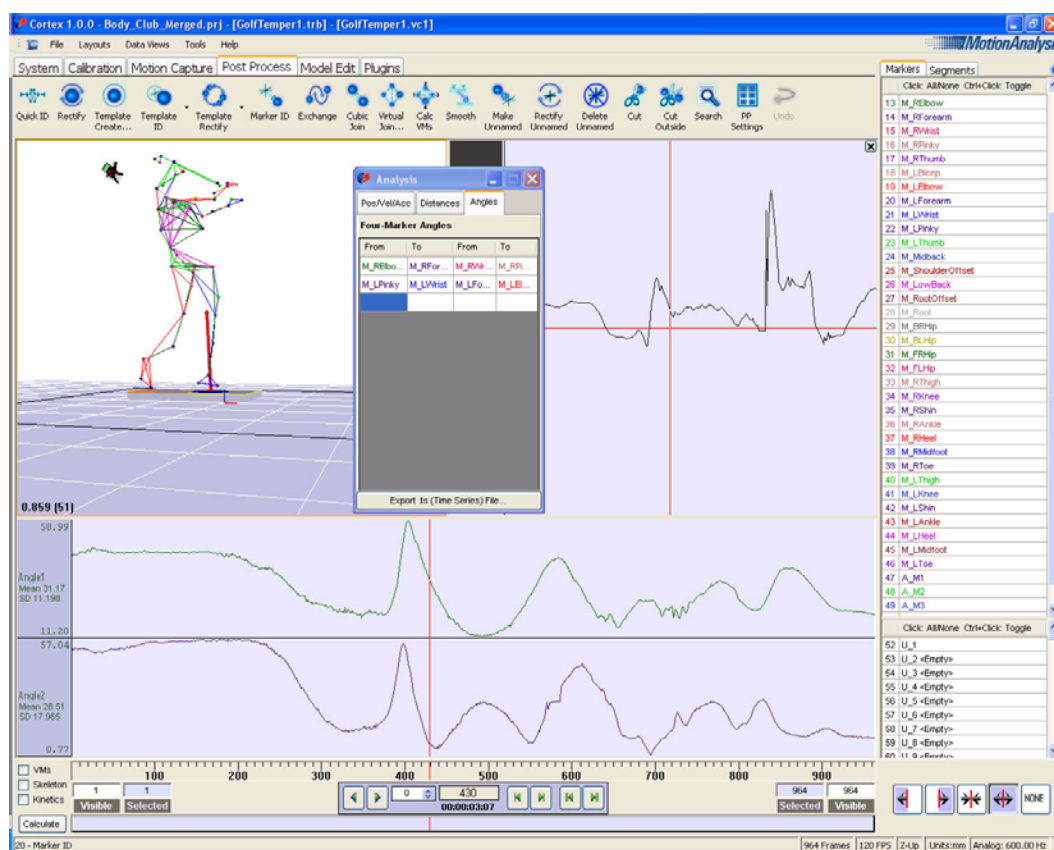
Figure 10-22. Distance Between Two Markers Tab



Included Angles Tab

The Included Angles tab allows you to select groups of four markers, which define two separate lines in space. Between these two lines, the angle is calculated for each frame through the tracked data. This information proves useful for detecting irregularities in movement, such as between two parts of a body. To delete a row, simply click on that row and press **Delete**. You may select and delete several rows at once by pressing **shift + click** on the rows and pressing **Delete**.

Figure 10-23. Included Angles Tab



Exporting Analysis (Time Series) Information

To export an ASCII text viewable file, select the **Export.ts (Time Series) File...** button. This will write a file with the same file name as the tracked file you are working on. It will contain the information from all three Analysis tabs.

Post Processing Strategies and Tips

The steps necessary to clean up data will vary significantly from one user to another. These tips are guidelines that outline a general approach to successful editing sessions.

- It is best to start identifying from Frame 1 forward. Then, identify from the last frame backwards. This entails naming unnamed markers using the Marker ID and Quick ID tools.
- Use of Rectify over small frame ranges may help in cleaning the data by taking unnamed data into named tracks.
- After identifying, the Post Process tools can be used to fill gaps and cut out unwanted data sections, fix abnormalities, and smooth anomalies, and can be used to exchange switched markers.

Note:

It is recommended to save your files often, especially when performing heavy edits.

- Many users will select all markers and all frames and execute a Join Linear or Join Cubic, and possibly a Smooth as a very last editing step.
- Learning and using Hot Keys is critical to high productivity.

Topic	Page
Overview	11-1
Markers Panel	11-2
Tree View Panel	11-4
Virtual Markers	11-17

Overview

The Model Edit tab provides tools to build and modify the model parameters that are mandatory for the project file. These parameters include markers, virtual markers, linkages, and segments.

Note: It is important to save your project after building the model by selecting **File > Save Project**. For more information about project files (*.prj), refer to [“PRJ—Cortex Project File” on page G-3](#).

Markers Panel

The Markers panel is intended for building and modifying marker sets.

Figure 11-1. Markers Panel

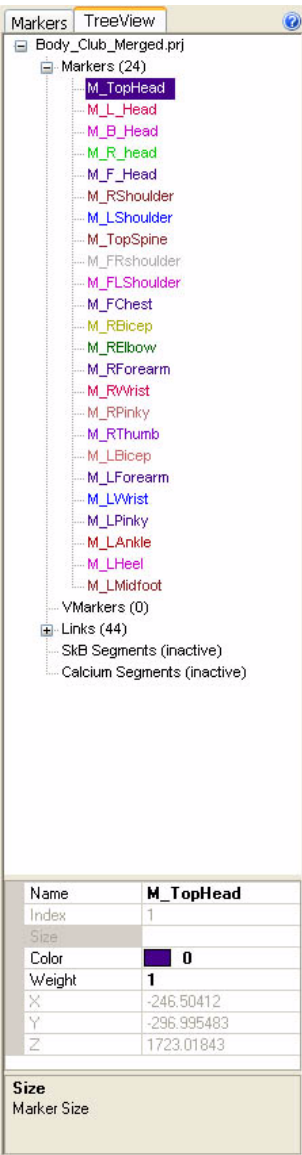


Clear Marker Set Button	This button clears out the project's marker set linkages and skeleton definitions
Create Linkages for Template Button	This button must be clicked prior to creating linkages on the 3D View. Linkages can be built by connecting the dots. Linkages should reflect the rigid or semi-rigid lines to aid in the template identifying mechanism.
Marker Names	Marker names are accepted in the markers grid when you press Enter on the keyboard.
Marker Color Selector	Allows the user to select a specific color for any marker. Select the marker from the list and simply click on the color of preference.

Tree View Panel

The Tree View panel provides an overview of the primary elements (markers, linkages, virtual markers, and segments) of the model and allows you to reorder the markers in the marker set by dragging and dropping. You can also insert and delete markers as well.

Figure 11-2. Tree View Panel



Project (.prj) File

Selecting the project file in the Treeview allows you to make changes to the current project file. The Project property values that can be changed include the following:

Name

This is the marker set name you can display over the marker cloud. The toggle function for this is found 3D View right-click menu. The default is set to the current project name.

Skeleton Engine

This provides a way to select which skeleton engine will be used to create the bone structure (if used). The choices are:

None

Turns off any skeleton calculations for the Motion Capture and Post Processing panels.

Skeleton Builder (SkB)

Turns on the skeleton builder definitions (if present in the project file) and allows the skeleton to be calculated. To see the skeleton in the Motion Capture or Post Processing panels, you must have the Show Skeleton feature turned on in the 3D view (right-click in the 3D view and select). All **Cortex** users can calculate the **SkB** skeletons after they have been defined, but creating and editing skeletons requires a separate software license.

Calcium Type Skeleton

Checking this option tells the software to use the Calcium Solver type skeleton. The version number of that software is listed after its name and it can be loaded independently with a different **solver.dll**. You must have the Show Skeleton feature turned on in the 3D window and you must have clicked the Calculate Skeleton "Bone Button" in Post Processing to have the skeleton calculated. Note that **SkB** skeletons are calculated automatically in Post Processing, but you must press the Big Bone button to calculate the Calcium Solver skeleton. To calculate the skeleton, you must also have a model file (the .mod file) of the same name as the project file in your current directory. Model files are created and edited in the **Si 2.0/Calcium** software. Any user can run and calculate the Calcium Solver skeletons, but it takes a separate license to edit and create the model files. The skeleton can be calculated in the Motion Capture panel from either live camera data or in the simulated realtime mode when you "Disconnect -Use Raw Files".

Note that this can have an optional MOD file associated to it.

SIMM OrthoTrak Skeleton

This skeleton calculation uses the Calcium Solver type skeleton and requires that you use the anatomically named marker set defined in the **SIMM** and **OrthoTrak** software. The **OrthoTrak SIMM** basic marker set uses fixed names like L.Shoulder, L.Wrist, and L.Knee for prominent marker locations. There are several required markers and many more optional markers. If you use this marker set, you do not need to use the **Si/Calcium** software for creating a model (.mod) file. You must also use an

additional "Static" trial with inside (medial) knee and ankle markers. You load the Static project file, load the Static Trial, click this **SIMM OrthoTrak** button and then the Big Bone (calculate skeleton) button becomes active. Then load the Walking (motion) trial and click the **Big Bone** button on the Post Processing screen. The Calcium Solver skeleton is calculated. To see the skeleton in the 3D window, select **Show Skeleton**. The skeleton can be calculated in the Motion Capture panel from either live camera data or in the simulated realtime mode when you "Disconnect -Use Raw Files".

Note that this requires an associated JNT file and an Init or Static trial.

Skin File

Skin files are rigid shells that do not scale with different sized subjects, and do not span/stretch across joints.

The Skin File function allows you to select one of four pre-defined skins that work with **Cortex**. These skins that are defined for two different skeleton types and are located in the **C:\Program Files\Motion Analysis\Cortex\User Files\Skins** directory. The four skin types are:

1. OrthoTrak Male (**MaleOTSkin.obj**)
2. OrthoTrak Female (**FemaleOTSkin.obj**)
3. OrthoTrak Polygons (**PolyBonesOT_Skin.obj**)
4. 25 Bone Animation Skin (**25_Bones_Male.obj**)

If you are looking to develop an entirely new skin file, you will need to contact Motion Analysis Customer Support (support@motionanalysis.com) for information.

Skin Transparency

This sets the transparency attribute of the skin—100% means the skin is invisible, 0% is solid.

Figure 11-3. Project Property Values

Property	Value
Name	Matt.prj
Skeleton Engine	Calcium Solver 1.2.4
Skin File	
Skin Transparency%	50

Showing the Skin

An OBJ skin consists of two files:

1. A skin file: **<Skin>.obj**
2. An associated base position: **<Skin>_Base.htr**

The HTR and OBJ file must have "group" names that match the base position HTR's segment names. The base position HTR file must have segment names that match the marker set. The order of the names doesn't matter. The matching is done with the currently selected skeleton engine.

To select it:

1. Go to **Model Edit >TreeView**.
2. Select the project name (first line of the tree).
3. Select the Skin File property (at the bottom of the sup-panel).
4. Select a file in the Open File dialog.

The filename is saved in the project file, so each marker set can include a skin.

Note: The skin file and the associated HTR file is not saved in the project file. Only their names and relative directory paths are saved.

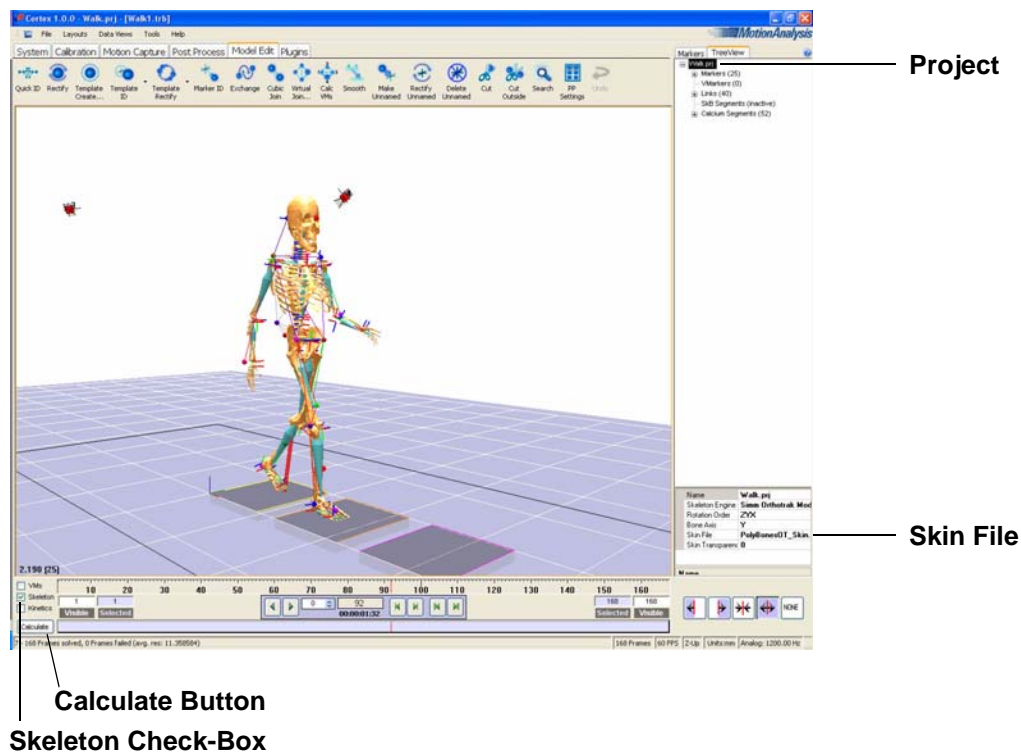
OrthoTrak Example

The following is an example of how to get the OrthoTrak skins to show in the 3D View.

1. Select **File > Load Project...** and load the project file **Walk.prj** located in the **Samples\Helen Hayes Markers** directory.
2. Select **File > Load Tracks...** and load **Walk1.trb**.
3. Go to the Model Edit tab and select the project (Walk) in the Tree-view.
4. Select the Skin File in Property Value and select **PolyBonesOT_Skin.obj**.
5. Select the **Skeleton** check-box in the lower-left of the **Cortex** interface and then click on the **Calculate** button directly below.
6. If not already set, right-click in the 3D View and select **Skin** and **Show**.

This procedure will produce a subject with a skeleton and skin as shown in Figure 11-4.

Figure 11-4. OrthoTrak Skeleton and Skin Subject



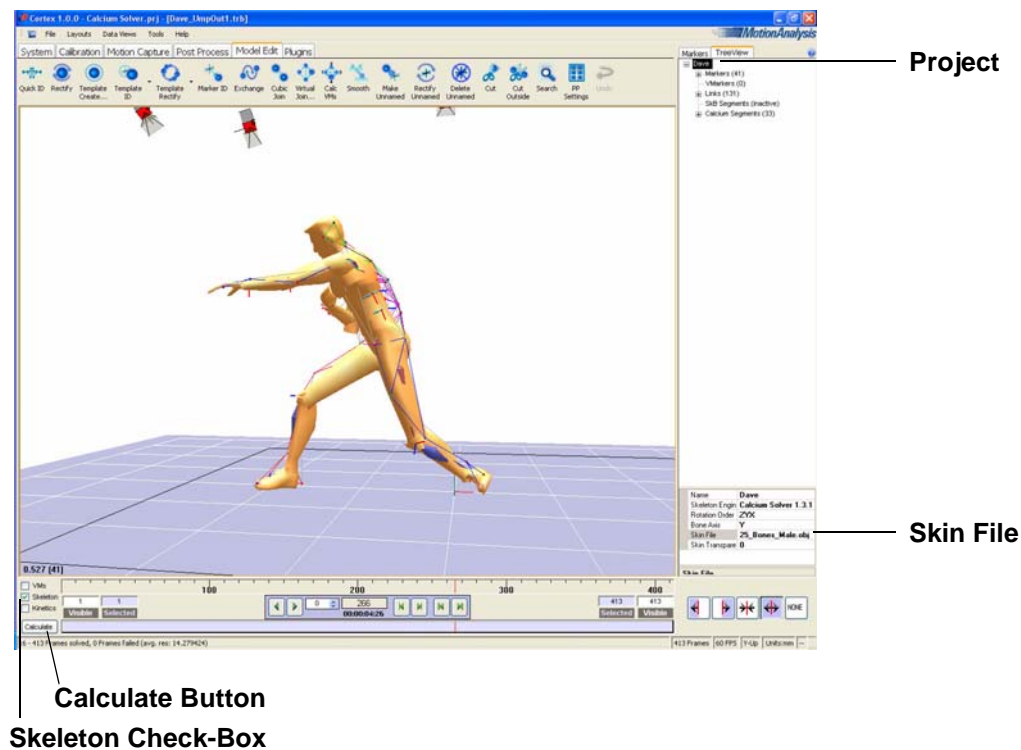
Animation Example

The following is an example of how to get the 25 Bone skin to show in the 3D View.

1. Select **File > Load Project...** and load the project file **Calcium Solver.prj** located in the **Samples\Talon Streaming Calcium and SkB** directory.
2. Select **File > Load Tracks...** and load **DaveUmpOut1.trb**.
3. Go to the Model Edit tab and select the project (Dave) in the Tree-view.
4. Click on the Skin File in Property Value and select **25_Bones_Male.obj**.
5. Select the **Skeleton** check-box in the lower-left of the **Cortex** interface and then click on the **Calculate** button directly below.
6. If not already set, right-click in the 3D View and select **Show Skins** and **Show Skeleton**.

This procedure will produce a subject with a skeleton and skin as shown in Figure 11-5.

Figure 11-5. Animation Skeleton and Skin Subject



Markers

Selecting Markers in the Treeview allows you to make changes to any of the markers associated with the current project file using the property value selections at the bottom of the panel. You can also insert, delete and select a range of markers. The marker values that can be changed include the following:

Name

Displays and edits the name of the selected marker.

Index

Displays the marker number, in the order the marker appears in the marker list. This is not editable.

Size

This is not used at this time.

Color

Displays and edits the color associated to the maker in the 3D View. To change the color, click on the color property and select from the drop-down menu.


Weight

This is not used at this time.

X, Y, and Z Values

The 3D coordinates of the marker in calibration units at the frame number where the marker is selected. It is not updated with every frame change, but is updated when you select the marker.

Figure 11-6. Marker Parameters

Name	M_RElbow
Index	13
Size	
Color	 9
Weight	1
X	-517.037
Y	-133.389023
Z	1087.20569

VMarkers

Selecting VMarkers in the Treeview allows you to make changes to the virtual markers in the current project file. The VMarkers property values that can be changed include the following:

Name	Displays and edits the name of the selected VMarker.
Index	Displays the VMarker number, in the order it appears in the list of VMarkers for the project. This is not editable.
Type	This provides a way to select which type of virtual marker to be created: <ol style="list-style-type: none"> 1. Two-Point (Ratio) 2. Two-Point (Value) 3. Three-Point (Ratio) 4. Three-Point (Value)
Origin Marker	Allows you to select and edit which marker is the Origin Marker of the Virtual Marker definition. To edit, click on the property and select from the drop-down menu.
Long Axis (Y)	Allows you to select and edit which marker is the Long Axis (Y) of the Virtual Marker definition. To edit, click on the property and select from the drop-down menu.
Plane Axis (XY)	Allows you to select and edit which marker is the Plane Axis (XY) of the Virtual Marker definition. To edit, click on the property and select from the drop-down menu.
X Offset	Sets the X coordinate of a VMarker definition.
Y Offset	Sets the Y coordinate of a VMarker definition.
Z Offset	Sets the Z coordinate of a VMarker definition.


Figure 11-7. VMarker Property Values

Property	Value
Name	V_LHip
Index	2
Type	Three-Point (Value)
Origin Marker	M_BLHip
Long Axis (Y)	M_FLHip
Plane Axis (XY)	M_Root
X Offset	50
Y Offset	70
Z Offset	-40

For more information on Virtual Markers, reference [“Virtual Markers” on page 11-17](#).

Links	Selecting any of the links in the Treeview allows you to make changes to the links in the current project file. The link property values that can be changed include the following: Right-click can only delete.
Index	Displays the link number, in the order the link appears in the list of links for the project. This is not editable.
Marker1	Allows you to select and edit which marker is the first end point of the link definition. To edit, click on the property and select from the drop-down menu.
Marker2	Allows you to select and edit which marker is the 2nd (of two) end point of the link definition. To edit, click on the property and select from the drop-down menu.
Color	Displays and edits the color associated to the link in the 3D View. To change the color, click on the color property and select from the drop-down menu.
Extra Stretch	The Extra Stretch factor is the amount (in percent, refer to Figure 11-8) that a link is allowed to stretch beyond the measured template. The template is usually measured from a Range of Motion capture and that motion is not always an adequate representation. Data captures often push the body closes to its limits. The Extra Stretch gives an appropriate margin to allow real captures to go beyond the template.

Figure 11-8. Link Property Values

Index	17
Marker1	M_TopSpine
Marker2	M_LShoulder
Color	 1
Extra Stretch	0%

What confuses the user most is that rigid body parts (the head is a good example) will not have their markers move around much no matter what data you use to create the template. So almost any template will work to give you values with high confidence. Conversely, if you have links that are stretchy, it's easy to not get a good template for it and so your confidence in the template will be less.

Because of this, many users think that the Extra Stretch factor is a statement about the elasticity of the link, which it isn't. A link could be put between a hand marker and the toe marker but as long as it had sufficient data for the template Extra Stretch could be set to 1 (the lowest) and have it work just fine.

Multiple Link Selection—Multiple links can be selected by pressing the **Ctrl + Shift** buttons on the keyboard and clicking on the individual links with the mouse. Color and Extra Stretch properties can be edited for setting groups of links together.

This is why there is a **CreateTemplate.sky** script. It shows how to use multiple TRB files as input to the template creation (and extension) process.

Assuming a standard Range of Motion (ROM) file, the Extra Stretch values are normally set as follows:

- 10 for the head
- 15 for the links on the hips/pelvis
- 15 for the feet
- 20 everywhere else

15 is the default value for new links.

This is a starting point. As data is tracked, you can fine-tune the Extra Stretch values. Smaller values that work consistently will result in fast identifying and fewer errors.

Create Template: Show Template Linkages

An effective tool in changing the settings and organizing the linkages can be found in the **Create Template > Show Template Linkages** check box under the Post Process Tool Strip. For more information, refer to [“Template Create...”](#) on page 10-13.

General Notes on Extra Stretch

- Extending the template works exactly as if you took all the TRB data used to create the template and put it end to end and did a Create Template with the whole works at once.
- If the ES factors are too large you get misidentified markers.
- If the ES factors are too small (and your template isn't complete enough) you will get unidentified markers.
- The template is a pair of Min/Max values for each link. These values only get farther apart as you extend the template. If this value becomes too great (as might happen if you had bad data to create the template), you must start the template creation process over from the beginning. There is no function to revalue the numbers, hence, this is why there is a Sky script for this. It makes it trivial to redo the template creation process.

SkB Segments

Name	Displays and edits the name of the selected SkB segment.
Index	Displays the SkB segment number, in the order the segment appears in the list for the project. This is not editable.
Parent	Displays the parent segment of the selected SkB segment. To edit, click on the property and select from the drop-down menu.
Origin Marker	Allows you to select and edit which marker is the Origin Marker of the SkB segment definition. To edit, click on the property and select from the drop-down menu.
Long Axis (Y)	Allows you to select and edit which marker is the Long Axis (Y) of the SkB segment definition. To edit, click on the property and select from the drop-down menu.
Plane Axis (XY)	Allows you to select and edit which marker is the Plane Axis (XY) of the SkB segment definition. To edit, click on the property and select from the drop-down menu.
RX Offset	RX is used to rotate the bone in the SkB segment along the X axis. RX is not used very often compared to RY. If you select a segment to rotate, it will bring up the rotation gizmo.
RY Offset	RY is used to rotate the bone along the Y axis. If you select a segment to rotate, it will bring up the rotation gizmo.
RZ Offset	<p>RZ is used to rotate the bone in the SkB segment along the Z axis. RZ is not used very often compared to RY. If you select a segment to rotate, it will bring up the rotation gizmo.</p> <p>For more information on Skeleton Builder, refer to the Skeleton Builder Quick-Start Guide found in the C:\Program Files\Motion Analysis\Cortex50\Samples\Skeleton Builder directory.</p>

Figure 11-9. Skeleton Builder Segments Property Values

Property	Value
Name	LFoot
Index	5
Parent	LLowerLeg
Origin Marker	V_LAnkle
Long Axis (Y)	V_LFoot
Plane Axis (XY)	M_LMidfoot
RX Offset	0
RY Offset	0
RZ Offset	0

The various skeleton types are described in Chapter 13, Skeleton Types.

Calcium Segments

Calcium is the graphical user interface to the Solver engine. Solver is the powerful numerical tool for calculating skeleton motion from marker data. The Calcium interface in **Cortex** is what allows you to correlate the positions of a marker pose to the initial pose of a skeleton. The skeleton is usually created in an outside animation package, such as Maya, 3D Studio Max or Kaydara and then exported to an HTR file by a Motion Analysis file IO plugin for that package. In this example we're using a skeleton from a Maya character.

Figure 11-10. Calcium Segments Property Values

Property	Value
Units	m
Global Scale	1
Matrix Method	Levenberg-Marquart
Accuracy	0.0001
Max Iterations	100
Use Joint Limits	True
Orient Body	True

Units

Units of the Calcium segment lengths. Select from the drop-down menu in meters, centimeters, and millimeters.

Global Scale

Changes the scale of the entire hierarchy, multiplied by the number set (e.g. a value of 10 would scale the hierarchy by 10 times the original size). This provides a quick method for scaling the HTR file to fit the marker cloud in the model pose.

Matrix Method

There are two matrix methods to choose from:

- **Gauss Newton**—Which is faster at solving, but not as robust. This is generally used in Real Time applications.

- **Levenberg-Marquart**—While is more robust, but not as fast. This should only be used in PP mode.

Accuracy

This is the accuracy parameter for the solve. It is generally set to 0.0001 and then left alone. It can be useful to debug and troubleshoot the Calcium segment.

Max Iterations

This is the number of iterations the solver goes through to solve. When the solver gets "stuck", it can potentially iterate forever. Usually the solve happens in a very small number of iterations (1-5). Setting it to 100 is more than enough.

Use Joint Limits

Enables or disables the use of joint limits in the Calcium model. Generally it is recommended to be turned off for animation applications. If the solution has joints flipping around, turn it off. Any model created from a joint file should have the limits enabled.

Orient Body

This is important on the first frame of any solve. First orient the root bone to the root bone markers, then do the solve. This helps to eliminate some first frame errors when bones get oriented incorrectly.

Note: There is still a bug where joints get turned around on the first frame. The solve changes randomly whenever these last two flags are changed. User beware.

It is recommended that Orient Body is set to False when Real Time operation performance is a factor.

For complete information on Calcium software and Calcium segment definitions, please reference the *Calcium for Cortex Quick-Start Guide* (p/n 651-1920-010).

Virtual Markers

Virtual markers are markers that get their position from a combination of the position of two or three actual markers in the motion capture data. Typically, a virtual marker is used to generate the actual joint center position of the performer (or subject) being motion captured. This is necessary since the actual markers lie on the outside of the performer. Joint center markers are desirable for use with analytical and skeleton reconstruction tools.

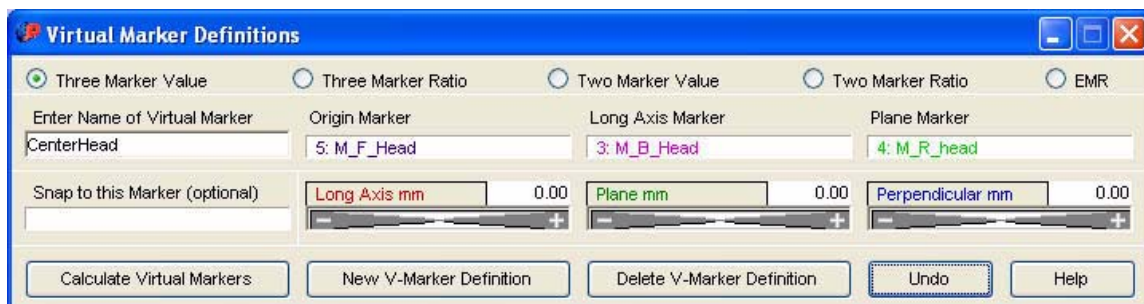
Virtual Marker Definitions

There are two methods for defining Virtual Markers (VM):

1. **2 Marker** - Two markers are used to define a line in space. A new virtual marker can be calculated anywhere on this line.
2. **3 Marker** - Three markers are used to define a plane in space. A new virtual marker can be calculated anywhere in space relative to the origin of this plane.

The placement of the virtual marker along a line or relative to a plane can be accomplished in real world measured values using the units of calibration as the units of measurement, or as a ratio. In the case of a line, the ratio is based on the distance between the two markers defining the line. In the case of the plane, the ratio is based on the distance between the two markers defining the Y axis of the plane.

Figure 11-11. Example Virtual Marker Setup



Types of Virtual Hierarchical Translation & Rotation Data

Currently, there are two primary methods of exporting the motion of a subject:

- ***.trc** (track row column) output files
- ***.htr2** (hierarchical translational rotation) output files

A ***.trc** file contains the X, Y, and Z translation values for the reflective markers relative to the capture volume's coordinate system. To translate this data into a hierarchical segment model requires software having an Inverse Kinematic (IK) Solver to create joint translations and rotations. This I.K. approach works well when the proportions of the subject are similar to the animation model.

An *.htr2 file contains hierarchical translation and rotation data representing the different identified segments (bones) of the subject's body. In this approach, you must establish virtual markers at the estimated joint centers and create segment coordinate systems for each segment. The virtual markers and segment coordinate systems are defined once for a particular marker set and then stored and recalled from the project file.

Degrees of Freedom

A segment's movement characteristics can be expressed as having various degrees-of-freedom. For example, if a single marker placed on the right shoulder is used to define the origin of the right upper arm, and we track this marker through space creating a trajectory, we will express the movement of the right upper arm origin as having 3 degrees-of-freedom (translations in X, Y, and Z).

If we add another marker to the right elbow and track it along with the marker on the right shoulder, we can now express movement of the right upper arm segment as having 5 degrees-of-freedom (translations in X, Y, and Z, and rotations in X and Z). This assumes that the Y axis extends from the right shoulder to the right elbow. If we add a third marker to the right wrist and track all three markers, we now have 6 degrees-of-freedom for the right upper arm segment.

Figure 11-12. Marker Number vs. Degrees of Freedom

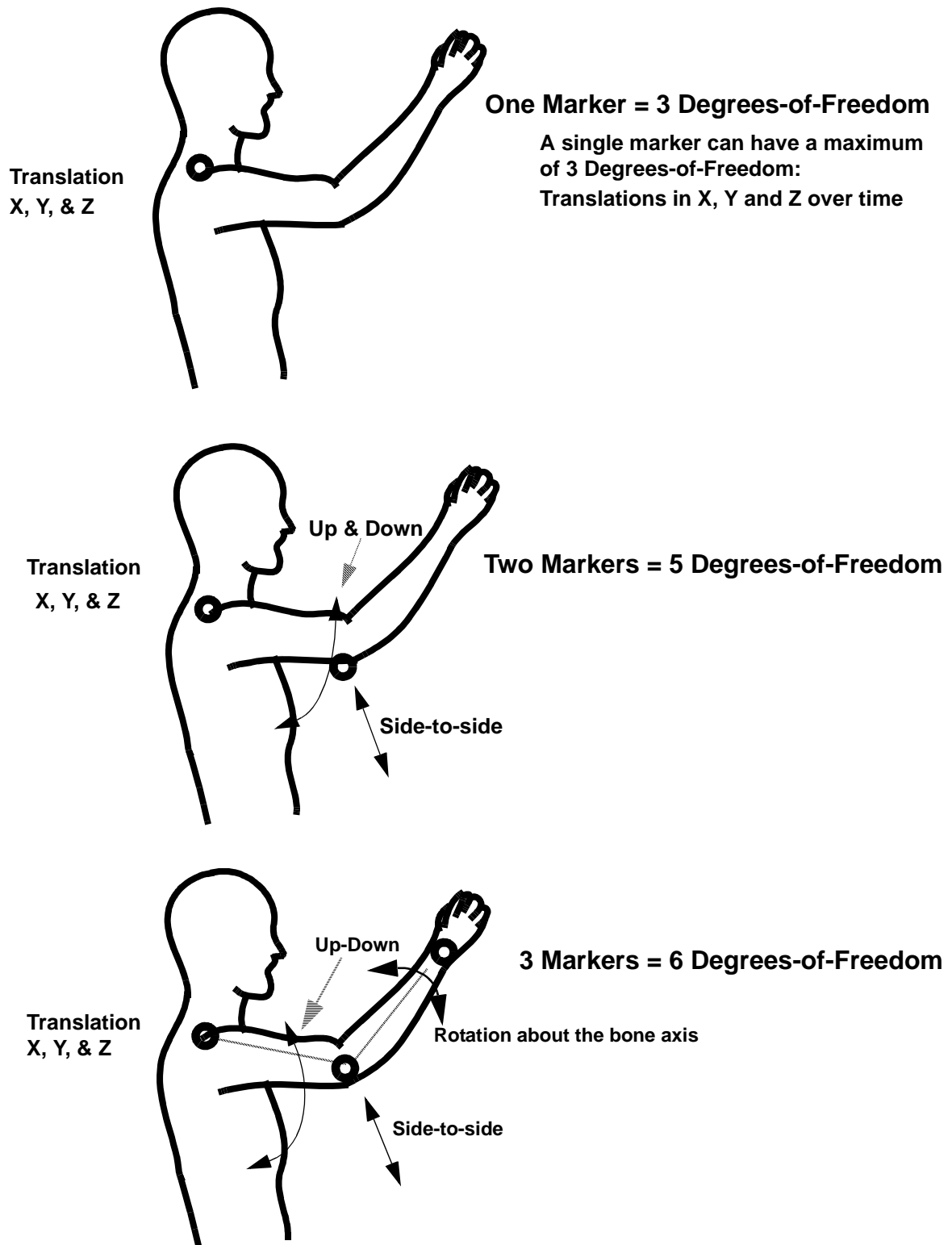
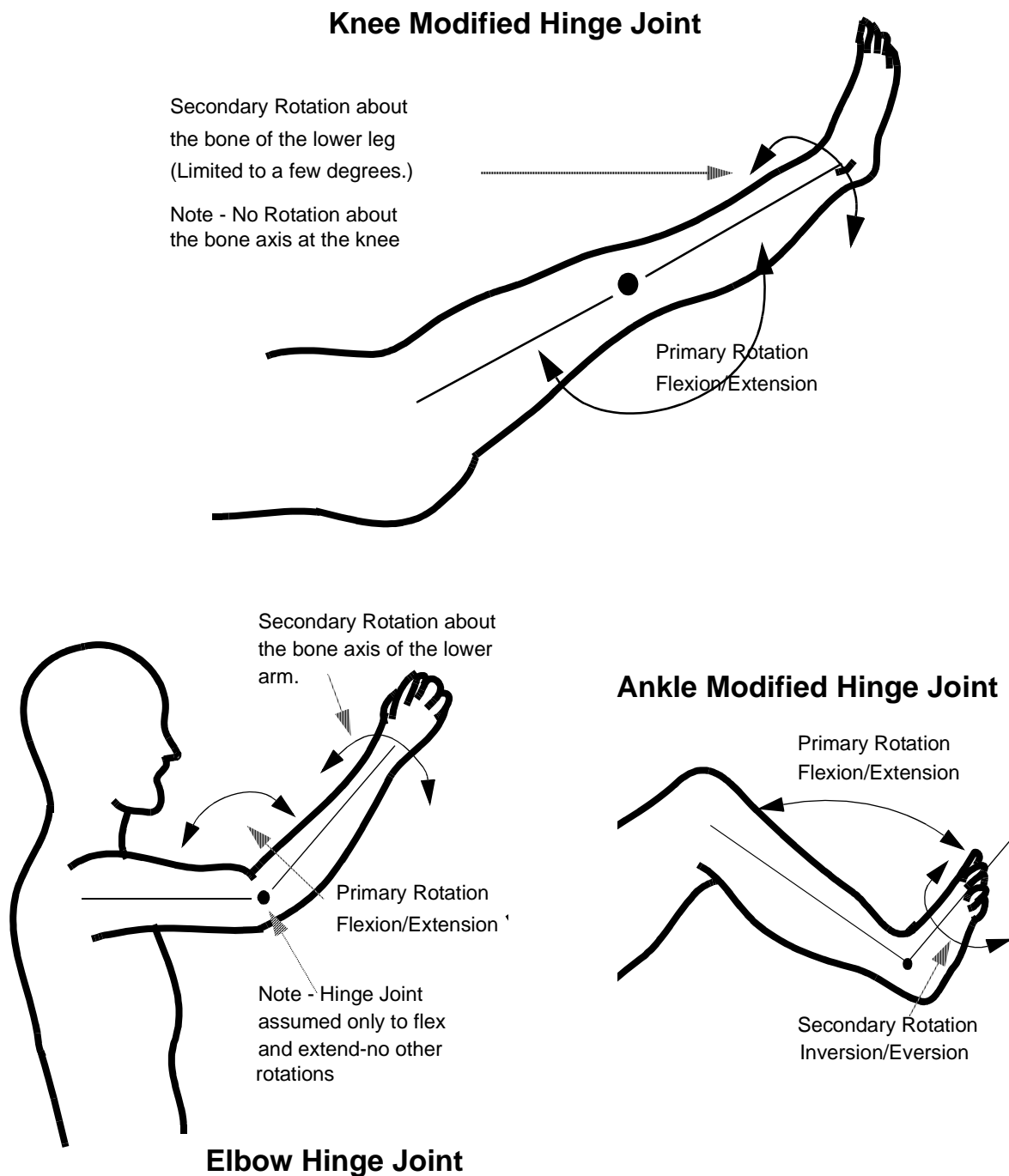


Figure 11-13. Examples of Hinge Joints



Calculating Virtual Marker Tracks

Calculate virtual marker trajectories based on the Virtual Marker definitions in the current project file. Virtual Markers are cleared when you track any new data.

Track and edit your trials before calculating the Virtual Marker Tracks.

1. Open the Virtual Markers Definitions form by clicking the **V-Marker Definitions** button in Model Edit or by selecting **Tools > Virtual Marker Definitions** in the Menu Bar.
2. Fill out the Virtual Marker Definitions form for the desired markers.
3. Click **Calculate** to calculate the virtual marker positions.

Virtual Marker Notes and Examples

For notes and examples on how to use Virtual Markers, there are two valuable sources:

1. Video tutorial from the Cortex Help menu: **Help > Tutorials > Cortex Skeleton Builder (SkB) and Kinetics Tutorials**
The first of this set of three tutorials shows how to set up and use Virtual Markers. To see this, you will need to install the full Samples and Tutorials from the Cortex installation CD or FTP site.
2. Cortex Kinetics User's Manual

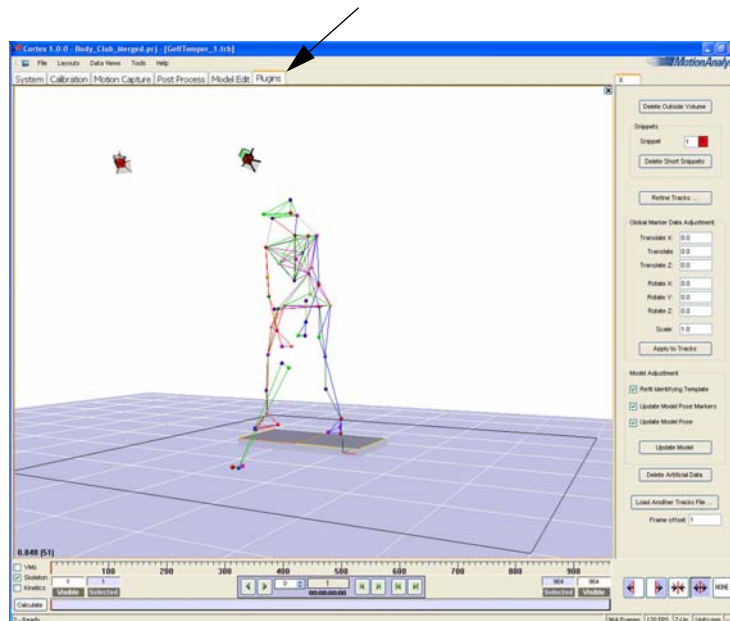
Chapter 12 Plugins Tab

Topic	Page
Overview	12-1
X Panel	12-2

Overview

The Plugins tab houses the X panel. Other plugins can be added by copying the **pluginname.dll** file into the **C:\Program Files\Motion Analysis\Cortex\Release1.x.x.x** folder and relaunching **Cortex**.

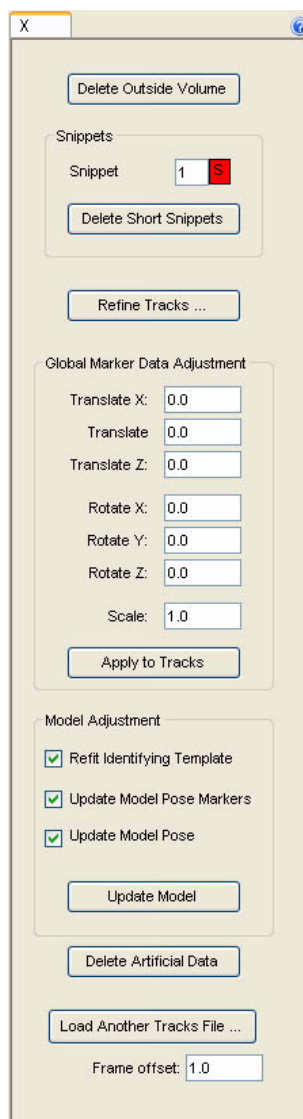
Figure 12-1. Plugins Tab



X Panel

The X panel provides a set of extra functions that are used in the post process mode.

Figure 12-2. X Panel



Delete Outside Volume

Eliminates all marker data outside of the volume defined in Calibration Details.

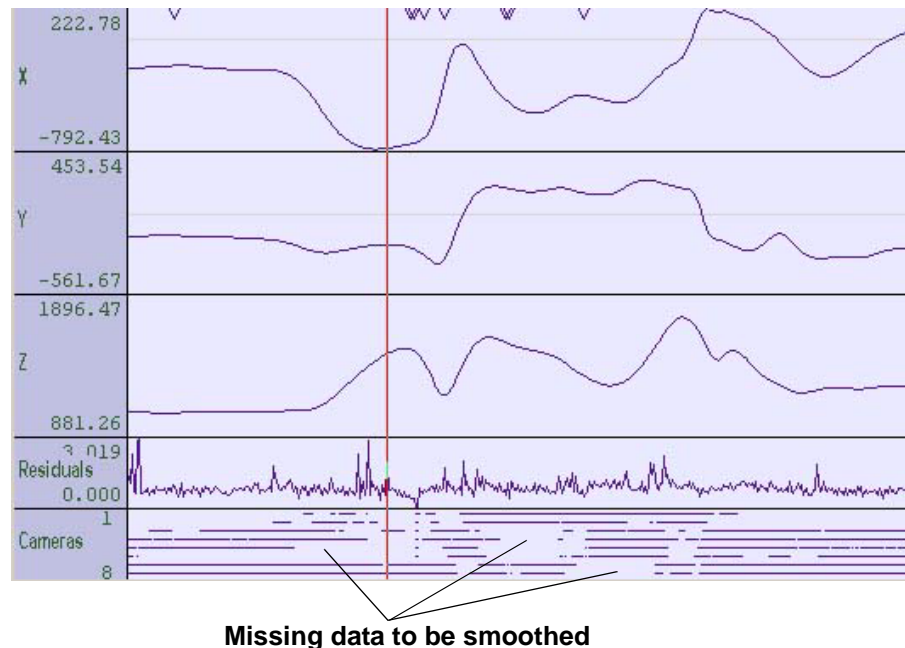
Snippets/Delete Short Snippets

Deletes data strings (in frames) that are shorter than a specified length.

Refine Tracks

This feature will smooth data that has become jumpy due to camera on/off noise. When a camera is turned on and off, there is at times a small data spike in the frames before the camera is turned off and after it is on. This is useful on facial data where small increments (< 1 mm) will have a significant effect on the final results (animated character).

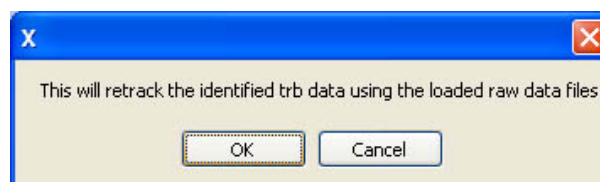
Figure 12-3. Refine Tracks



To Refine Tracks you will need to:

1. Load a *.prj file.
2. Load a *.trb file that has been tracked and identified. Only identified tracks are refined. Unnamed marker data is not affected.
3. Select **X Panel > Refine Tracks...** The message appears as shown in [Figure 12-4](#).

Figure 12-4. Retracking Identified TRB Data Message



4. Select **OK**.
5. At this point, the Load Raw Video File interface opens. Select and open the VC files that were used to create the tracks.
6. Select **File > Save Tracks**.

Global Marker Data Adjustments

The Global Marker Data Adjustment section allows the user to modify the tracks data by translating, rotating, and/or scaling the data. This is an operation that applies to all marker data over all the tracks. It's especially handy for converting the overall orientation of the data (such as from a Z-up coordinate system to a Y-up coordinate system).

Model Adjustment

The Model Adjustment section allows the user to update the Calcium Solver model pose data and the template model pose data simultaneously (this is the data displayed when the Show Model Pose flag is on). The marker data in the model pose is used for two different operations: as a starting pose for the template when doing a New Subject operation; and as the matching base pose for the skeleton in the Calcium Solver model. Doing both adjustments at once is important to maintaining the integrity of the data. None the less, the user is allowed to change them independently if necessary.

Refit Identifying Template

This option takes the current frame in Post Process and compares the template linkage lengths of that frame with the stored model pose. The template Min/Max values are re-calculated based on the amount of change in the linkage lengths.

Update Model Pose Markers

This check box indicates that the stored marker model pose is to be replaced with the marker data on the current frame.

Update Model Pose

This check box indicates that the stored skeleton model pose is to be replaced with the current skeleton data that has been calculated for the current frame.

These last two options are used to update the model pose of a performer between motion capture sessions (such as from one day to the next). This avoids having to spend time refitting the position of the skeleton to the new day's model pose data. The user should still verify that the fit is a good one, but if the markers on the performer have not moved by very much then the fit is likely to be good.

Update Model

The updates the model using the check-boxes as set above in the panel.

Delete Artificial Data

Deletes the Virtual Join, Cubic Join, Linear Joint, etc. data that was created for the current tracks file.

Load Another Tracks File

This merges another track trial into memory so you can view two different data trials in the 3D window. A Frame Offset lets you offset in time the merged data. Only positive offsets are allowed. It is meant for visualization of multiple data sets in the 3D window. The first trial that is loaded must have enough frames to wholly accommodate the additional trials as

additional memory for any additional frames is not allocated and can cause unpredictable results. You can make one trial that has enough frames to fully accommodate all of the merged data frames and load that track data set first with the **File > Load Tracks** menu item. The merged data does not get marker names or linkages unless you have a project and marker sets defined for the additional tracks.

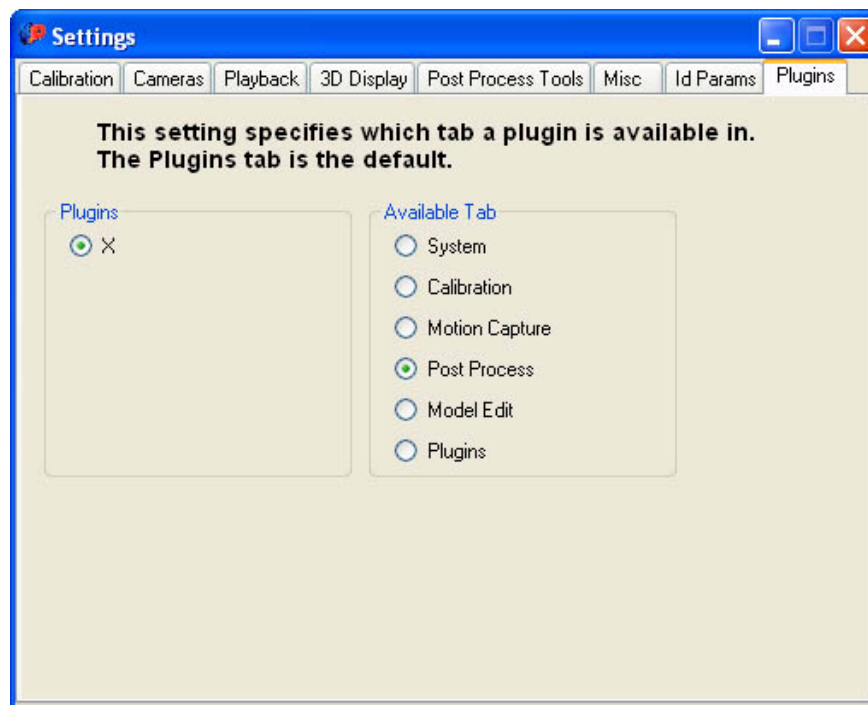
Frame Offset

This offsets the first tracks file with respect to the second. The first tracks file loaded must have enough frames to accommodate all additional tracks files. If not, unpredictable results may occur. This is intended for viewing multiple trials. To see the stick figures, multiple marker sets must exist in your current marker set.

Moving the X Panel to a new Tab

You can move the X panel to a different tab (System, Calibration, Motion Capture, etc.) with the **Settings > Plugins** tab. This provides a custom function that may help improve workflow.

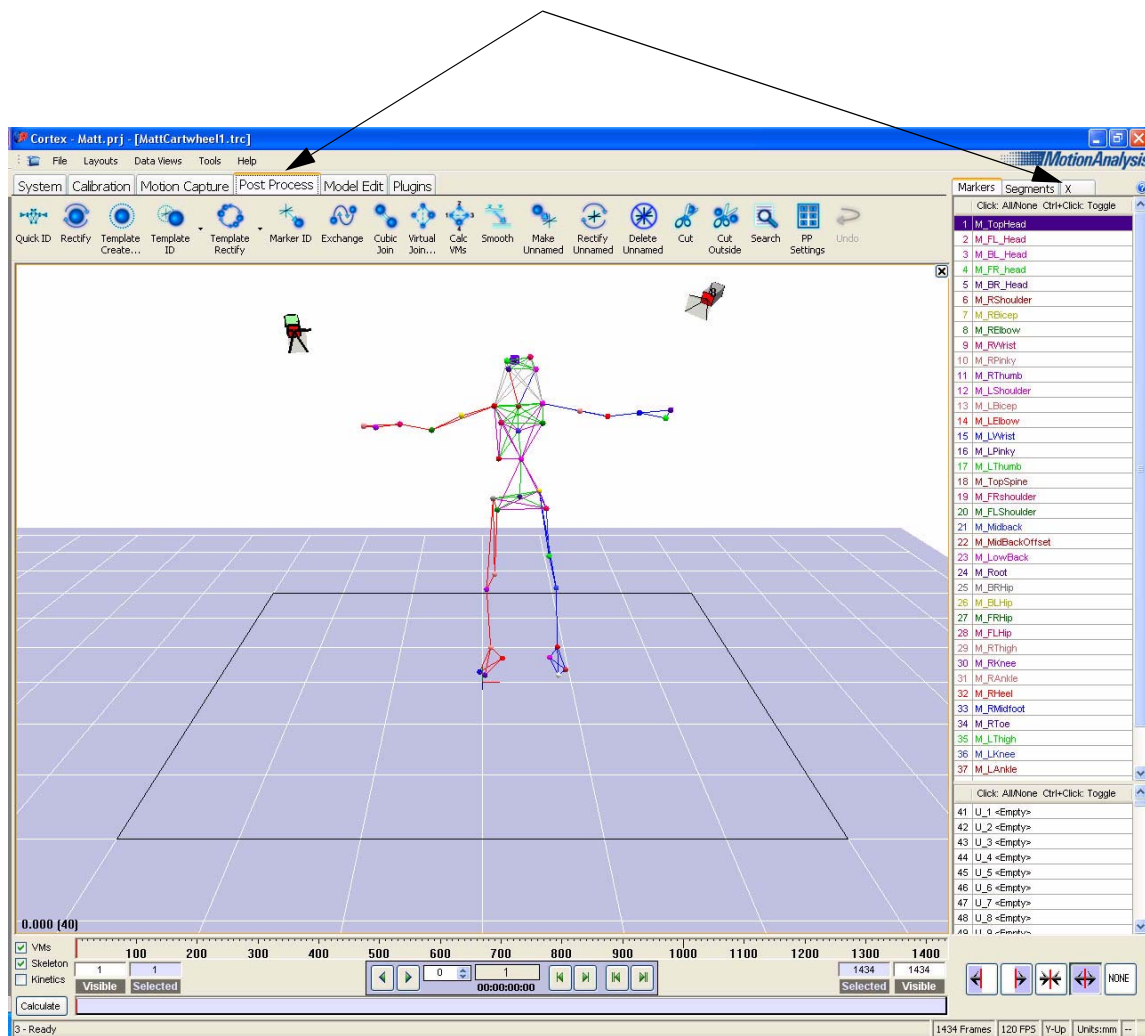
Figure 12-5. Settings > Plugins Tab



In this example, the X panel is moved to the Post Process tab (see [Figure 12-6](#)).

Figure 12-6. X Panel in Post Process Tab

The X Panel is now located in the Post Process Tab

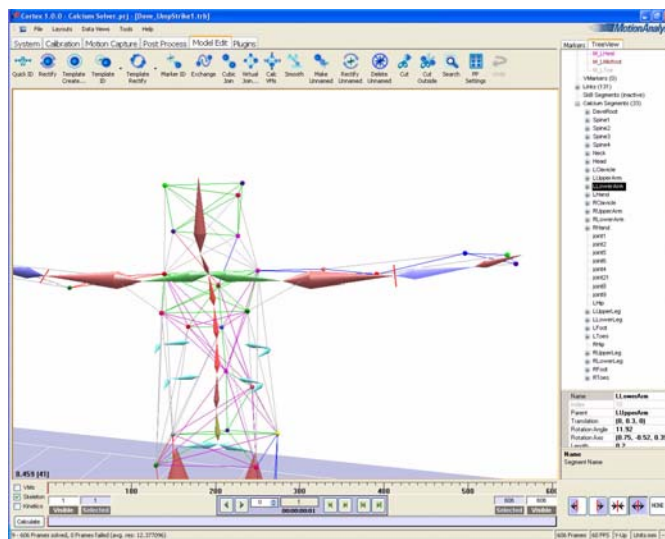


Topic	Page
Overview	13-1
Skeleton Builder (SkB) Skeletons	13-2
Calcium (Solver) Skeletons	13-2
Which Skeleton Engine Should I Use?	13-3
Cortex Skeleton Engine Selection	13-3
Skeleton Option Details (as selected from Tree View)	13-5
Exporting the Skeleton Data Into an HTR File	13-7
Multiple Characters and Multiple Skeletons	13-8

Overview

Cortex supports two kinds of skeleton calculations: **Skeleton Builder (SkB)** skeletons and **Calcium Solver** type skeletons. Either kind is calculated in the **Cortex** software and either can be calculated from live camera data, simulated Real Time with VC files, or from XYZ data in Post Processing. Both the marker data and the skeleton data are available to the **Talon** streaming plugins, such as the Maya and Kaydara Talon streaming plugins. The user can write their own plugin with the **Talon SDK** (Software Development Kit) available from Motion Analysis.

Figure 13-1. Calcium Solver Integrated with Cortex



Skeleton Builder (SkB) Skeletons

Skeleton Builder (SkB) skeletons are relatively simple, direct and fast calculations of segments (bones) that are defined and calculated from one marker center to another. The markers can be real or virtual (calculated) and are typically from one virtual joint center to a second virtual joint center. A 3D local coordinate system is defined with 3 markers:

1. One marker defines the origin.
2. A second marker defines the bone axis.
3. The third marker defines the plane.

The advantage of the SkB type skeleton is that they compute very quickly and they represent the method of how most biomechanical models have been computed for many years. The disadvantage is seen when viewed on a skinned character in an animation package: The bones (segments) change length as a result of the calculation method. This is due primarily to the motion of the markers on the skin which change from frame to frame. An animated character can be set up so that the character keeps a fixed length skeleton and the skeleton is driven only by the angles measured from the skeleton. This has the visually undesirable artifact that the character's feet will appear to slide on the floor and possibly raise above or protrude below the floor.

Calcium (Solver) Skeletons

Solver type skeletons are calculated quite differently than SkB skeletons. Solver uses the Global Optimization method of calculating the best fit of the marker data to the underlying fixed bone length skeleton. This technology was pioneered by Motion Analysis in 1990. The early version model setup was somewhat cumbersome and required physically measuring from a person's joint centers to the marker locations before the skeleton could be used. Now, the skeletons are defined and edited within the Calcium software interface within **Cortex**. This provides a graphical way of either reading in an existing skeleton (or creating one) from an animation package such as Maya or 3DMAX. The typical way would be to create the character in the animation package and export the skeleton using an htr file using the MAC File IO plugins. The Global Optimization method is an iterative method of seeking the best fitting of the skeleton within the "marker cloud" of identified markers. The results are quite astounding: the animated characters motions derived from this method is very good. Final editing of the htr skeleton motion data can be done within the animation package or with a third party tool, such as Kaydara's Motionbuilder.

Which Skeleton Engine Should I Use?

SkB skeletons are good for most biomechanical applications and have been the norm there for many years. They are also used for animation customers who want tools for quick pre-visualizations of your characters motions.

For the final cut and the big screen presentations, where details matter and looks are everything, you will be glad to have your Calcium (Solver) skeletons under the skin of your final characters.

Cortex Skeleton Engine Selection

The Engine Selection sets the method of calculating the (optional) skeleton model that can be calculated in **Cortex**. There are two main skeleton engines available from Motion Analysis: Skeleton Builder and Calcium (Solver). **Cortex** will run any skeleton engine that has been previously defined without any additional licenses. To define or edit the skeleton definitions, you need the Skeleton Builder or Calcium license items.

The skeleton model is saved with your marker set information in the project file. The **File > Load Marker Set** menu item will load the skeleton type from a PRJ file once it has been stored there.

Post Process Skeletons

Skeletons can also be calculated in the Post Processing tab from your current XYZ data that is visible in the 3D and XYZ data views.

SkB Skeleton Builder Skeletons

Skeleton Builder (SkB) skeletons are always calculated if the **SkB Skeleton Builder** option is selected. You can see the Skeleton by selecting **Show Skeleton** from the right mouse menu in the 3D data view. If you are using one of the streaming plugins, like the Maya Talon plugin or the Motionbuilder plugin, both the marker data and the calculated skeleton is available to drive your animated character from either the Real Time or Post Processing mode.

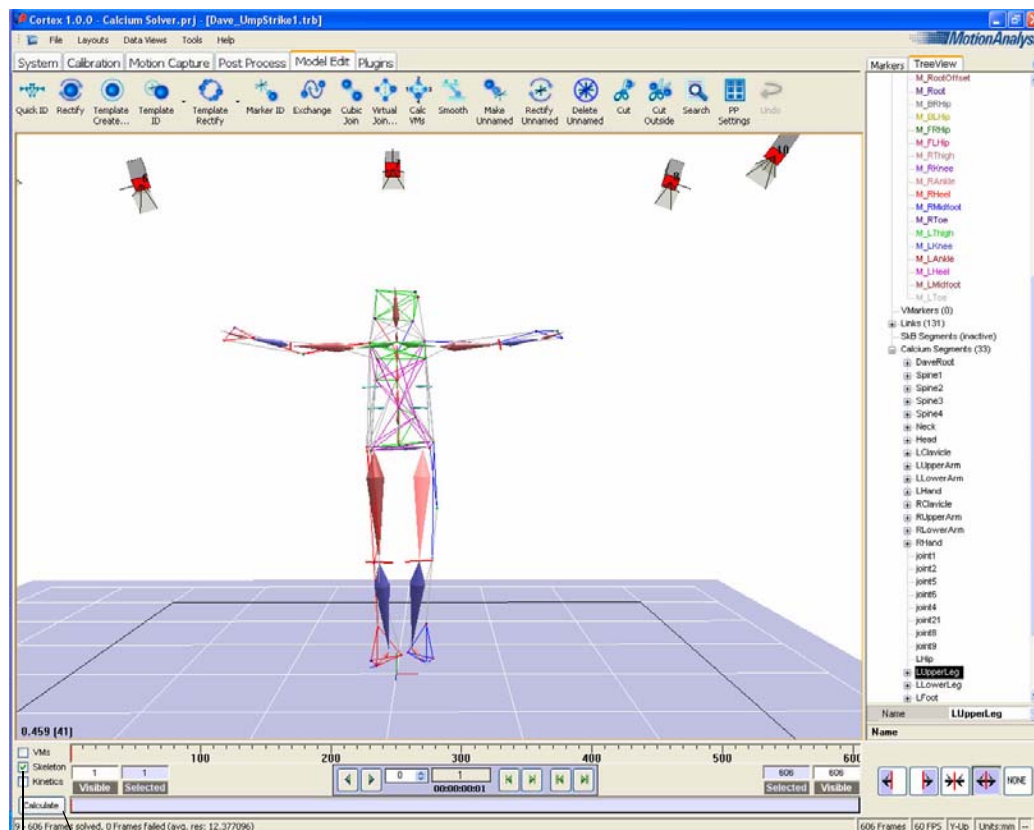
Calcium Solver Skeletons

The Calculate button (refer to [Figure 13-2](#)) activates the Solver engine to do the Global Optimization method and solve for the skeleton within the marker cloud. If you have defined a Calcium Solver Skeleton in Post Processing, the process is as follows:

1. Select the Skeleton check-box.
2. Define a period of frames (select a starting and end frame) to calculate from the current time slider.
3. Click on the **Calculate** button.

Note: The Treeview has an option to view the Skeleton as it renders.

Figure 13-2. Skeleton Check-Box and Calculate Button



Calculate Button

Skeleton Check-Box

This can take several seconds to minutes, depending on the length and complexity of the skeleton. If you are using one of the streaming Talon plugins, the skeleton data is available after the calculations are finished. You can scrub back and forth in the Post Processing mode or press the **Play** button and both the marker data and the skeleton (htr type) data are available to the streaming Talon plugins (like Maya).

Skeleton Option Details (as selected from Tree View)

Skeleton Builder (SkB) Engine

Bone segments are defined from one marker to another, typically virtual markers that represent joint centers. The advantage to this method is the fast and direct calculations from markers to joint centers to bone segments. The disadvantage is primarily for high resolution animation use since the joint centers are calculated directly from real and virtual marker locations. Bone lengths will vary slightly from frame to frame due to marker-skin motion which can cause the animated character's skin to distort and not look as good as expected.

Calcium Solver 1.3.1

This is a very different method of calculating the skeleton motion from marker locations. Typically, the skeleton is defined within one of the several animation packages and exported and saved in an HTR and a MOD (model) file. This skeleton is not allowed to change size to fit the motion data, but the Solver engine software uses a best fit Global Optimization of the marker data to conform to the rigid underlying skeleton. This results in the very best way of animating characters from mocap data, but to use it in **Cortex**, you need to save a .mod file with the same name as your prj file. The **Calcium** software allows you to export a .mod file. The Solver Global Optimization method is resident in three of Motion Analysis Corporation's software products: **Cortex**, **Calcium**, and the **SIMM** modeling package.

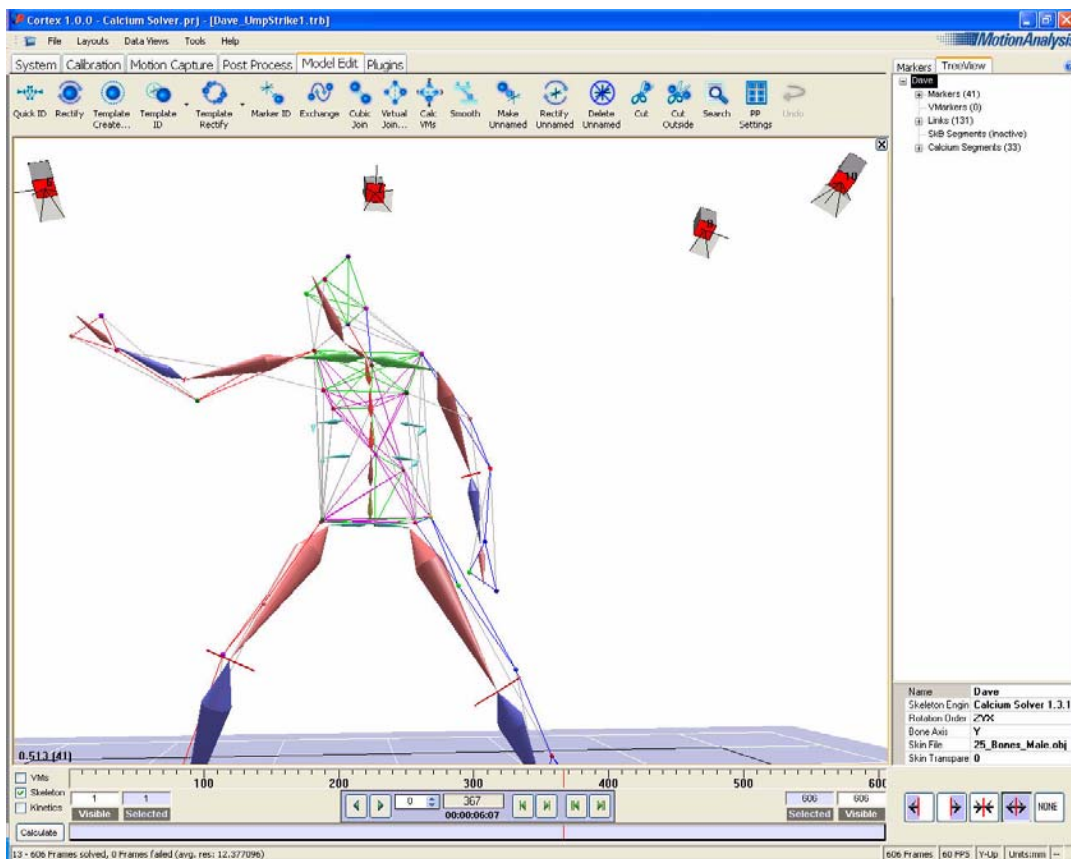
SIMM Calcium Model

This uses the Solver engine with the same advantages as the **Calcium** Solver method above, but with a known and fixed marker set that was developed for biomechanics use. To use it, you must use some variation of the **OrthoTrak** marker set, which has several required markers (such as the Knee, Ankle, Hip and Shoulder markers) and many optional markers that will introduce more detail and more bone segments into the solution. The big advantage over the **Calcium** Solver model is that you do not need to create a MOD file, which means that you do not need to use or learn the **Calcium** software. **Cortex** creates an even more thorough JNT (joint) file when you export the SIMM Calcium Model. You do need to have the person standing in a neutral pose (typically with the arms out in a T-pose, feet slightly apart and thumbs forward).

Model Edit > TreeView

The Skeleton Engine type is displayed and can be edited in the **Model Edit > TreeView** panel when you select the prj name at the root of the Tree View. The Skeleton Engine type appears as the Value of the Skeleton Engine Property at the bottom of the Tree View panel.

Figure 13-3. Model Edit TreeView



Exporting the Skeleton Data Into an HTR File

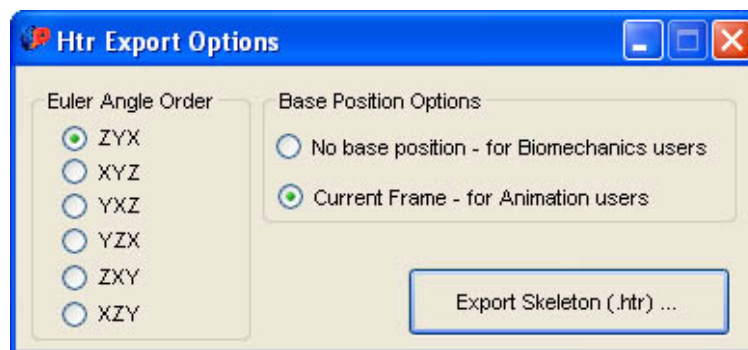
The **SkB** or Solver type skeleton data can be saved to an HTR (hierarchical translations and rotations) file after you calculate it and view it in **Cortex**. Select the **File > Export HTR file...** menu item. This is for use with animation packages. You will then select the default top Euler Angle Order (ZYX) since that is how the plugins are built to receive the data.

The following are options on the Export HTR file menu:

Euler Angle Order

Use ZYX (which is the default) if you are going to import this with a Motion Analysis File IO plugin to the animation packages. Other Euler Angle orders will be decided by your local mathematicians. The numbers in the **Cortex** software are stored internally in a certain way and exported to the HTR file according to the method above.

Figure 13-4. HTR Export Options



Base Position Options

Angles in the columns of HTR data are absolute angles according to the coordinate systems defined. This is typically used by biomechanics and research customers.

Current Frame

The absolute angles in the currently selected frame are written out in the file header of the HTR file. The angles in the columns are zero referenced to the angles in the file header. Use this method option if you are going to read the htr files into an animation package with the Motion Analysis File IO plugins.

Licensing Notes

SkB skeletons are defined and edited within the **Cortex** software and require a separate license item, but they can be run and the skeletons data generated without additional licenses in **Cortex**. There is separate documentation provided with the Skeleton Builder that shows how to set up and edit **SkB** skeletons.

Calcium Solver skeletons are imported or created in the **Calcium** software. **Calcium** software requires a separate license to edit or create the skeletons, but they too can be run and the skeleton data created from **Cortex** without a separate license. **Calcium** also can create the HTR skeleton data using the same Solver engine as **Cortex** and the **SIMM Motion Module**.

Multiple Characters and Multiple Skeletons

When you specify additional marker sets, the skeleton engine needs to be in each of the project files that you select. The skeleton type is stored in the project file. For previously stored project files, open up each of the project files separately, **Tree View** panel, and specify the appropriate skeleton engine, then save out the project file. Load one of the project files, then in the **Motion Capture > Objects** panel, specify the second project file as an "Additional Tracking Object". You should then be able to load a .vc file, and both skeletons should become solved.

Topic	Page
Overview	14-1
Installation File Structure	14-2
The Registry	14-2
The Script Object	14-2
Script Examples	14-2
Updating Old Sky Files	14-2
Graphical User Interface	14-5

Overview

Sky is the name of the scripting interface for **Cortex**. It uses the VB Script engine to provide the semantic structure of the language along with Visual Basic to provide the graphical user interface of the window pane. Sky is intended as a tool for users to encapsulate elements of repetitive tasks such as file processing, data editing, and parameter setting. This tool is intended for users who have some general knowledge of scripting and programming.

Most of the Sky functions are direct, simple wrappers for the corresponding **Cortex** calls. Some exceptions have to do with sending messages to **Cortex** and in re-arranging arrays of data that get passed back and forth.

Installation File Structure

Sky is integrated with **Cortex** and uses the **Cortex UserFiles** folder to store scripts and documentation.

- **UserFiles/SkyFiles**—Contains the list of Global Sky functions. This is the root of the Sky folder structure.
- **UserFiles/SkyFiles/CopyPerProject**—This folder contains the Sky files (which you are encouraged to add to) that will get copied to any current working folder using the “CopyPerProject” toolbar button.
- **UserFiles/SkyFiles/SkyDocs**—Contains the Sky documentation files. The root folder contains this document while the **SkyFunctions** sub-folder contains an RTF file for each Sky function.

The Registry

Sky stores two keys in the Windows Registry under **HKEY_LOCAL_MACHINE\SOFTWARE\Motion Analysis\Sky**. The first key is "Network Folder" and saves the path for the Network Sky Scripts list box. The second key is "StartupScript" and saves the filename of the Sky script to run when Sky is opened for the first time.

The Script Object

The scripting language is VB Script. Sky loads the script into the VB Script object which compiles then executes the script code. You can create and use variables, subroutines and functions in VB Script. These values persist from one Sky script to the next so you can set values in one script and use the results in another script.

Script Examples

Sample scripts with data are found under the **Cortex** installation folder under **Samples\SkyExamples**.

Updating Old Sky Files

The Sky scripting language for **Cortex** is very similar to the one used for **EVaRT**, but there are a few small differences based on how Sky invokes the VB Script object. The most important difference is that the "Main()" subroutine declaration in a Sky script is no longer required. In fact it is recommended that you not use it any more (although Sky will correctly run a script with such a declaration). The easiest way to update an old script is to rename the "Main" subroutine to something appropriate, such as the name of the Sky script. Then add a line to the script to call the subroutine.

For example:

```
'
' FindAutoIDFrame - Old version
'
' Increment frames and keep calling AutoID until a
marker gets identified.
' This assumes no markers were identified before being
called.
'
sub Main
    swPopups_SetQuietMode(1)
    iHi = swLoadedTracks_GetNumberOfFrames() -1
    for i = 0 to iHi
        swLoadedTracks_SetFrameNumber(i)
        swPost_AutoID()
        iResult =
swContext_GetMarkerPosition(0,x,y,z)
        if iResult = 0 then
            Message( "AutoID found at frame " +
CStr(i))
            exit sub
        end if
        if swCancelled() then
            Exit Sub
        end if
    next
    swPopups_SetQuietMode(0)
end sub
```

Would be modified to look like this:

```
'
' FindAutoIDFrame - New Version
'
' Increment frames and keep calling AutoID until a
marker gets identified.
' This assumes no markers were identified before being
called.
'
FindAutoIDFrame()
sub FindAutoIDFrame()
    swPopups_SetQuietMode(1)
    iHi = swLoadedTracks_GetNumberOfFrames() -1
    for i = 0 to iHi
        swLoadedTracks_SetFrameNumber(i)
        swPost_AutoID()
        iResult =
swContext_GetMarkerPosition(0,x,y,z)
        if iResult = 0 then
            Message( "AutoID found at frame " +
CStr(i))
            exit sub
        end if
        if swCancelled() then
            Exit Sub
        end if
    next
    swPopups_SetQuietMode(0)
end sub
```

The advantage of this technique is that the function is guaranteed to run the exact same way. However, it could also be modified to remove the function call completely and just execute the operations in-line. For example:

```
'  
' FindAutoIDFrame  
'  
' Increment frames and keep calling AutoID until a  
marker gets identified.  
' This assumes no markers were identified before being  
called.  
'  
swPopups_SetQuietMode(1)  
iHi = swLoadedTracks_GetNumberOfFrames() -1  
for i = 0 to iHi  
    swLoadedTracks_SetFrameNumber(i)  
    swPost_AutoID()  
    iResult = swContext_GetMarkerPosition(0,x,y,z)  
    if iResult = 0 then  
        Message( "AutoID found at frame " +  
CStr(i))  
        exit for  
    end if  
    if swCancelled() then  
        Exit for  
    end if  
next  
swPopups_SetQuietMode(0)
```

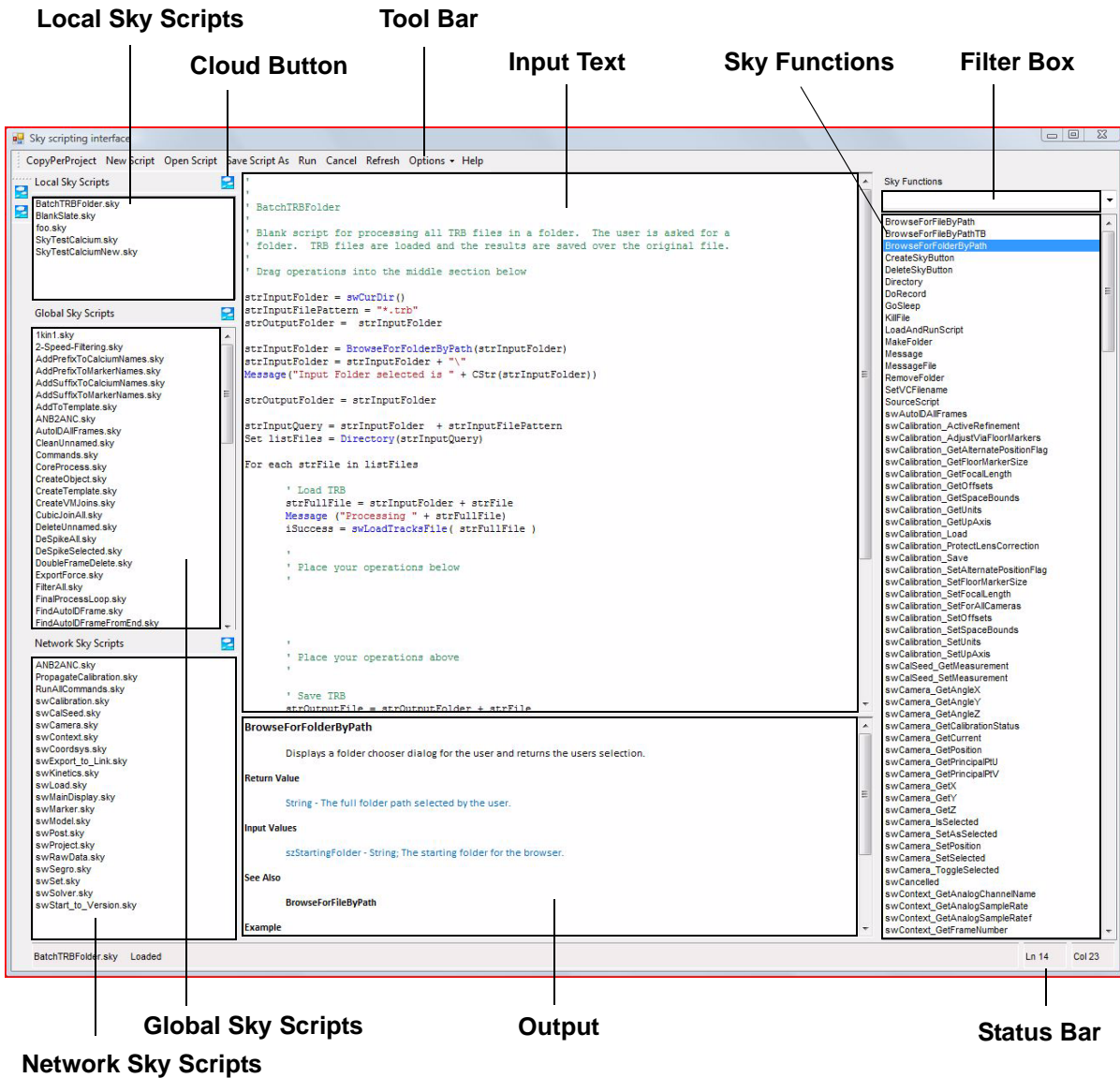
Note the difference in how the swCancelled function is used.

Not all the Sky functions that were in the EvaRT are in Cortex. But all the same functionality exists so you should be able to update the script with no difficulty by finding the replacement function. Most of these changes were in the Solver functions – instead of getting individual values of X, Y and Z with 3 different functions those 3 functions were replaced with one call that gets all 3 values, swSolver_GetSegmentJointAxis, for example.

Graphical User Interface

The user interface for Sky is found under the **Cortex** Tools menu.

Figure 14-1. Sky Graphical User Interface in Cortex



Toolbar

- **Copy Per Project**—Copies Sky files from the folder **MAC_DIR/UserFiles/SkyFiles/CopyPerProject** to the current working project directory. This makes it easy to initialize a new project with your favorite scripts. This will not overwrite existing scripts of the same name.
- **New Script**—Clears the script text editor and starts editing a new script.
- **Open Script**—Brings up a file browser to find a Sky file not in the current directory.
- **Save Script As**—Saves the script text to a new Sky file.
- **Run**—Executes the text in the script editor.
- **Cancel**—Cancels the currently executing script. This works only if the script was written to use the `swCancelled` Function.
- **Refresh**—Causes Sky to refresh the **Local Sky Files, Global Sky Files, Network Sky Files** and clears the Input Text area by re-reading the appropriate directory.
- **Options**
 - **Script Edit Mode**—Enables editing in the Input Text window. To run the script the **Run** button must be selected (or the menu short cut **Ctrl-R**. This is the default mode. This mode is the opposite of "One Touch Execution" mode.
 - **One Touch Execution**—Disables the editing in the Input Text window. In this mode Sky scripts are loaded and run immediately when selected from any of the Sky file lists.
 - **Set Network Folder...**—Sets the folder location for the "Network Sky Scripts" file list. This value is saved in the registry.
 - **Autosave**—When on (checked) this mode causes the Sky interface to auto-save the currently selected Sky file when ever the Sky file is changed, refreshed, or run. This is the default mode. When off, the file is not saved at any time so the user must use "Save Script As" to save the file.
 - **Reset Sky Script Object**—This clears the VB Script object of all variables and functions so it is just like when the Sky interface was first opened.
 - **Colorize Text**—When on (checked) the text in the Input Text window is colorized. This is the default. When this flag is off, no colorization is set. All Sky functions are shown in blue, comments are shown in green, and strings are shown in red.
 - **Set Startup Script...**—Specifies the startup script used by Sky. This script is automatically run whenever Sky is first opened. This is where Sky buttons are saved so they can be regenerated when Sky is opened.
 - **Save Startup Script**—Saves the current state of the Sky interface to the specified startup script. Currently this only saves Sky button information.
 - **Clear Startup Script**—Deletes the startup script.
 - **Run Startup Script**—Runs the startup script.
- **Help**—Brings up the help information in Internet Explorer

Local Sky Scripts

This contains a list of all the Sky files in the same directory as the current project. Load a file into the script interface by single clicking on the file-name. When you click a new file name, any changes you made to the currently loaded file are saved automatically (if the Autosave flag is set under the Options menu). The currently loaded file name will continue to be highlighted for as long as that file is current.

The "Cloud" button to the right of the Local Sky Files label brings up the Windows Explorer for the local folder. This makes for easy access to all the standard Windows file system browsing and editing tools.

Global Sky Scripts

This contains a list of all the Sky files in a standard, global directory for all the users on that particular computer. This directory is a sub-folder of the Userfiles folder under the Cortex installation folder. Load a file into the script interface by single clicking on the filename. When you click a new file name, any changes you made to the currently loaded file are saved automatically (if the Autosave flag is set under the Options menu). The currently loaded file name will continue to be highlighted for as long as that file is current.

The "Cloud" button to the right of the Global Sky Files label brings up the Windows Explorer for the global folder. This makes for easy access to all the standard Windows file system browsing and editing tools.

Network Sky Scripts

This contains a list of all the Sky files in a common network directory. This location is intended to be a shared network folder accessed by users on different computers. When you click a new file name, any changes you made to the currently loaded file are saved automatically (if the Autosave flag is set under the Options menu). The currently loaded file name will continue to be highlighted for as long as that file is current.

The "Cloud" button to the right of Network Sky Files label brings up the Windows Explorer for the network folder. This makes for easy access to all the standard Windows file system browsing and editing tools.

Input Text

The Input Text area displays the currently loaded script. This is what will be run when the Run button is selected.

Right-clicking in the Input Text window brings up a menu of editing tools:

- **Undo**—One level of undo is available in the Input Text window
- **Cut**—Standard text operation
- **Copy**—Standard text operation
- **Paste**—Standard text operation
- **Delete**—Standard text operation
- **Run**—Runs the script (Same as the Run button in the Toolbar)
- **Run Highlighted**—Executes the highlighted text
- **New Script**—Same as "New Script" operation in the Toolbar
- **Open Script**—Same as "Open Script" operation in the Toolbar
- **Script Save As...**—Same as "Save Script As" operation in the Toolbar

Output

Displays output from the scripting engine. Error messages from the scripting engine will be displayed as well as any text messages from the script. The output text area is refreshed when a new file is loaded. All help information for Sky functions is displayed here as well.

Sky Functions

The scripting functions (for the most part) in Sky are simple wrappers for functions found in the **Cortex** dll. Any call you make is immediately passed on to the corresponding core function in **Cortex**. These are the same functions that the Cortex GUI uses at runtime so you can do just about anything through the scripting interface that **Cortex** does interactively.

All the script functions have the same name as the corresponding dll function call except a "sw" has been added to the beginning ("sw" stands for "Sky Writing"). Online help for all functions is found from the Sky Functions list on the right (also, clicking on a highlighted Sky function inside the Input Text window will display help for a function).

The Sky functions filter box will display only those Sky functions which contain the text in the filter. This makes it easy to search for functions that contain a particular name such as "Camera" or "Marker" or "Load". A drop down menu retains previously typed in filters. Only those names shown in the Sky Functions list are highlighted in the Input Text window.

The Sky Functions list is generated automatically from the internal scripting objects so the list is guaranteed to display all available Sky functions.

Status Bar

The status bar at the bottom of the window displays messages regarding operations going on in the Sky interface. The Line and Column fields display the current location of the cursor. They are convenient for locating the line in a script causing a syntax or other error in the VB Script engine.

Drag And Drop

Drag and drop features are implemented in a number of areas in the Sky interface:

- Dragging and dropping a Sky file onto the Input Text area automatically generates the Sky command that will run the dragged script when the current script is run.
- Dragging and dropping a Sky file onto the toolbar creates a Sky button that, when selected, will execute the script that was dragged onto the toolbar. These buttons will be saved in the Startup script (see the Options menu description above) when you select "Save Startup Script".
- Dragging and dropping a Sky function name from the list on the right onto the Input Text area will drop the function name onto the script.
- Dragging and dropping a Sky file from one list to another will copy that file from the original location to the dragged one. If a script of that name already exists then nothing happens.
- Text from the output area (especially sample Sky commands from the Example section) can be dragged onto the Input Text area.

Appendix A *System Hardware Interconnections*

Topic	Page
Overview	A-1
Standard Eagle and Hawk System Configuration	A-2
Power Consumption	A-3
CP-8 Power Hub and Ethernet Switch Connections	A-5
Eagle and Hawk Digital Camera Connections	A-6
Network Interface Cards and IP Addresses	A-7
Configuring Ringlight Changes for Eagle and Hawk Cameras	A-8

Overview

This chapter provides information and illustrations on how to set up the hardware to be used with **Cortex**.

When using **Cortex** in the Motion Analysis motion capture system, hardware connections are straight-forward. The connection of the cameras to the CP-8 Power Hub and Ethernet Switch all have unique labeled connectors.

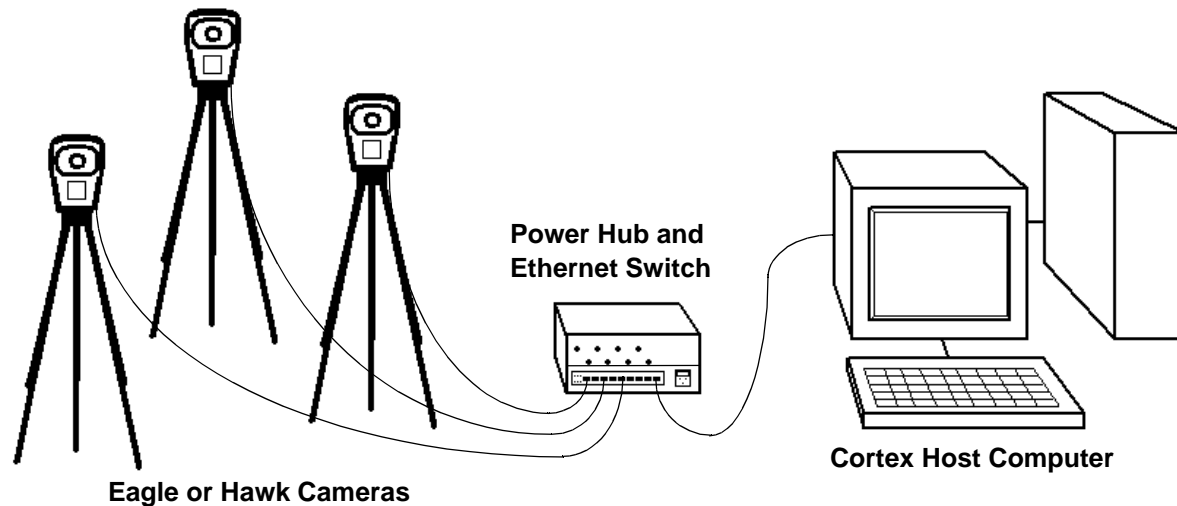
Note: Frame rate, shutter speeds, and ring light brightness for the Eagle and Hawk digital cameras are set using the **Cortex** user interface.

Note: When connecting the camera cables to the hubs/switches, you must use the end with the red indicator.

Standard Eagle and Hawk System Configuration

Figure A-1 shows a standard **Cortex** system setup for use with Eagle and Hawk digital cameras.

Figure A-1. Standard Eagle or Hawk System Configuration



For more detailed diagrams, please refer to Figures A-3 through A-10.

The main components include:

- a set of Eagle or Hawk digital cameras with Eagle network and power cables for each camera
- a CP-8 Power Hub (1 for every 8 cameras) and Ethernet Switch (1 for every 15 cameras), see [Figure A-2](#)
- a Tracking Computer (host) with monitor, keyboard, and mouse

Figure A-2. CP-8 Camera Power Hub and Ethernet Switch



Power Consumption

The maximum power consumption you can expect for example Eagle and Hawk system configurations is shown in [Table A-1](#). The actual power consumed depends on the video frame rate and the intensity of the ring lights and is usually less than that indicated.

For the most reliable system operation, it is recommended that all camera assemblies and computers be powered by an uninterrupted power supply (UPS). If you want to save your data when power is lost altogether, you will also need to power the tracking computer monitor from the UPS.

Table A-1. Power Consumption of Typical Eagle and Hawk Systems

	8 Cameras	12 Cameras	16 Cameras
Power Hub/E-Net Switch with Cameras	265 W	400 W	530 W
21" SVGA Monitor	125 W	125 W	125 W
Dual Processor Computer	200 W	200 W	200 W
TOTAL	580 W	725 W	855 W

Basics of Ethernet Switches and Hubs

There is a difference between the older Ethernet hubs and the newer Ethernet switches, even though they look alike and are functionally similar. The difference is in the performance. Ethernet switches guarantee the full rated Ethernet bandwidth between all ports simultaneously, whereas the older Ethernet hubs share the bandwidth for all ports. We call the EagleHubs "hubs" to indicate that they are the center connection point for a block of cameras, but inside the EagleHub resides a switch. This performance difference is important and necessary for the Eagle and Hawk systems to function properly.

A 100 MB Ethernet switch works well for an Eagle or Hawk system with up to 15 cameras at a capture frame rate from 60 to 120 Hz. For larger numbers of cameras (above 16), it is important to use the 1 GB Ethernet NIC (network interface card) inside the computer and the 1 GB Ethernet switch that collects and concentrates the camera traffic to the **Cortex** Host computer.

Ethernet Tutorial and Troubleshooting Guide

There are two types of female Ethernet connections that use the same 8-pin Ethernet connector:

1. NIC-Type—The Ethernet connector that is found on the Network Interface Card on computers and on your Eagle or Hawk cameras.
2. Hub-Type—The standard Ethernet plug that is found on Ethernet switches and hubs.

There are two types of Ethernet cables:

1. Standard Ethernet patch cables
2. Ethernet cross-over cables

The standard patch cable is used to connect computers to Ethernet switches.

The Ethernet cross-over cable is used for connecting Ethernet switches to other switches, unless you use the Uplink port on either switch. In this case, you can use a standard patch cable. The cross-over cable would also be necessary if you were to bypass the Ethernet switch and plug the Eagle or Hawk camera directly into your computer's NIC.

Troubleshooting

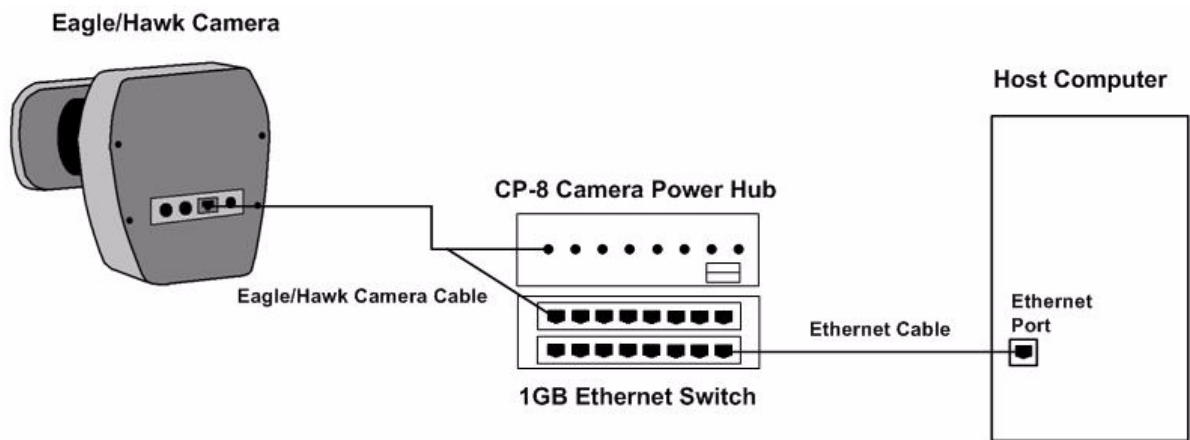
You can tell if you have a live Ethernet connection if the indicator light goes on when you plug the cable into the hub. This is also the best way to figure out whether or not your Ethernet cables are plugged in correctly. It will not damage anything if you plug in the wrong type of cable (patch or cross-over) into an Ethernet jack. For the indicator light to go on, there has to be a live Ethernet connection on both 'ends of the cables.

CP-8 Power Hub and Ethernet Switch Connections

When first setting up your Eagle or Hawk system, you will notice that both the power and Ethernet connections for the cameras are integrated through the CP-8 Power Hub and a 1 GB Ethernet switch. Remember, the Eagle or Hawk system allows for 8 cameras per CP-8 Power Hub.

All camera power connectors are plugged into the power connectors of the CP-8 Power Hub. Order is not imperative as long as each power connector is close to an open Ethernet connector on the 1 GB Ethernet switch.

Figure A-3. Standard Digital Camera Configuration (1 to 8 Cameras)



Eagle and Hawk Digital Camera Connections

Eagle or Hawk digital cameras are connected to the EagleHubs using camera cables with both power and Ethernet connectors. When fitting connectors together, be sure the connections are secure and snap firmly into place.

Figure A-4. Eagle Rear Panel Connectors



Power Connector	The power connector powers the camera with a 48 Vdc source from the EagleHubs, through a CAT5 cable.
Ethernet Connector	The Ethernet connector is set for a 4-wire, full duplex 100 Mbps Ethernet.
Aux Connector	The Aux connector can be used for testing VGA and diagnostics and is generally not needed for normal customer use. An Aux cable is supplied with each Eagle and Hawk system which has three connectors on one end (VGA, COM 1, and BNC) for use in various applications and diagnostic testing.
Future Connector	The Future connector is reserved for future use.

Network Interface Cards and IP Addresses

It is recommended that an Eagle or Hawk system have two Network Interface Cards (NIC):

1. Single port for your own LAN connection
2. 4-Port dedicated to the Eagle or Hawk system

Configuring a Network with Your Eagle Cameras

The IP addressing for the camera system is initially setup in the following manner.

- Digital Network Interface Card (NIC) = 10.1.1.199
- Subnet Mask = 255.255.255.0

This will need to be set in the “Camera Network IP Address” box in the **System > Cameras** panel.

Note: The NIC address is set from the Windows Control Panel. Select **Network Connections > Local Area Connection**, right-click **Properties > Connection** and use the following items:

- Internet Protocol (TCP/IP), Use the following IP Address: 10.1.1.199
- Subnet Mask 255.255.255.0
- DNS Server settings do not matter
- Other addresses will also work as long as you are consistent with the camera addressing and do not overlap IP addresses.

Ethernet Camera Addressing

The default camera addresses as configured when they are built start at 10.1.1.201 for camera number 1 with ascending numbers. For example:

- Camera 1: 10.1.1.201
- Camera 2: 10.1.1.202
- Camera 3: 10.1.1.203

All have sub-net masks of 255.255.255.0.

It is not necessary for the cameras to have ascending numbers. When you Connect to Cameras in **Cortex**, the software polls the network camera network and reports the number and the kinds of digital cameras.

Configuring Ringlight Changes for Eagle and Hawk Cameras

If you change the ringlight type on your Eagle or Hawk camera, you must tell the camera what kind of ringlight it currently has attached. If you do not, it could send too much power to the ringlight and damage the electronic circuitry or LEDs.

Parts You Will Need

1. Computer with COM port and HyperTerminal software (standard with Windows).
2. Eagle/Hawk Test Cable (shown below) that came with your Eagle or Hawk system. This cable is about 6 feet in length and has a special connector on one end which plugs into the AUX port on the back of the camera. The other end has three connectors: a 9 pin COM port, a 15 pin VGA, and single BNC.

Figure A-5. Eagle/Hawk Test Cable



Steps

1. Put the camera on a table, connect the single connector end of the Eagle/Hawk Test cable to the back of the camera (AUX port).
2. Connect the Eagle/Hawk Test cable's COM port to the COM1 port of your computer. COM2 may also be used.
3. Launch HyperTerminal. If you have installed **Cortex**, there should be a file named **EagleCOM1.ht** in the folder containing the **Cortex** executable (for example: **C:\Program Files\Motion Analysis\Cortex\EagleCOM1.ht**). Double click on the HT file name and HyperTerminal should launch. If you are on a different computer, look from the Start Menu under **Programs > Accessories > Communications >**

HyperTerminal. This allows you to view and type messages to the software in the Eagle or Hawk camera. You must quit **Cortex** (or uncheck **Motion Capture > Output: > Enable External Trigger** to free up the COM1 port. COM port settings: 9600 8-N-1.

4. Boot-up the camera by plugging it into the CP-8 Power Hub or turning the power off and on to the CP-8 Power Hub. After a few seconds you should see messages similar to the following example:

```

* * * * *
*  MAC Camera Control Program
*  Camera Configuration version 1.4  copyright
    (c) 2003, Motion Analysis, Inc.
*  NET+WORKS Version 3.00  copyright (c) 2000,
    NETsilicon, Inc.
* * * * *

Serial channel used for diagnostics will use a
baud rate of 9600.

After the camera board is reset, the camera
will wait 5 seconds for the user to signal any
changes on the keyboard.

- - - - -

Press any key within 5 seconds to change these
settings.
```

5. Press the Space bar (or any other key) to change the camera configuration.
6. Press **M** to modify the camera settings
7. Press **Enter** about 5 times to leave the other settings unchanged. Wait for the message that tells which Ringlight is currently configured. If the ringlight type is infrared, then the display should be **Ringlight type = infrared. (7)**. If the ringlight type is red, then the display should be **Ringlight type = red. (1)**. For example, to change from: near Infrared (4) to red (1), type in the 1 character as:
Ringlight type = near infrared. (4) 1
(You type in the single digit 1 followed by Enter)
8. Press **Enter** to leave the other items unchanged.
9. Wait for the camera to boot again and check that Ringlight type was successfully changed.

Appendix B *Analog Input Hardware and Software*

Topic	Page
Overview	B-1
Installing NIDAQ Software on an Cortex Computer	B-3
Analog Signal Naming Conventions	B-4
32-Channel, 16-Bit NI USB-6218 Configuration	B-5
NI USB-6259 Analog Channel Connections	B-13

Overview

The **Cortex** system can accept analog data from external devices and synchronize it with video motion data.

Analog cards known to work with **Cortex** software include the following A-D configurations and the necessary NIDAQ software. **Cortex** will support one or two of the devices listed in [Table B-1](#). The devices must have the same resolution (12-Bit or 16-Bit). Other NI Analog A-D input configurations should work but have not been tested.:

Table B-1. A-D Configurations Used with Cortex Software

A-D Configuration	NIDAQ Software
NI USB-6218, 32-Channel, 16-Bit (up to 6 USB devices, up to 192 channels)	NIDAQ 8.3 or later, EVaRT 5.0.4 or later NIDAQ 8.5 or later for Windows [®] Vista
NI PCI-6071E, 64-Channel, 12-Bit	Traditional NIDAQ 7.0 > 7.4, EVaRT 4.4
NI PCI-6071E, 64-Channel, 12-Bit (higher performance)	NIDAQ MX 8.0 or later, EVaRT 5.0
NI DAQ Card-6024E, 16-Channel, 12-Bit	Traditional NIDAQ 7.4, EVaRT 4.6
NI PCI 6254, 32-Channel, 16-Bit (up to 2 cards)	NIDAQ MX 8.0 or later, EVaRT 5.0
NI USB-6259, 32-Channel, 16-Bit (up to 2 USB devices)	NIDAQ 8.1 or later, EVaRT 5.0.2

Note: If you do not find your NI device in listed in [Table B-1 on page B-1](#), you may need to reference older versions of the **EVaRT** User's Manual.

Note: If you are using NIDAQ MX 8.0 or 8.1 versions and if you are collecting data for only one channel in **Cortex**, you will need to install a jumper wire from screw terminals PFI 7 to PFI 0 on the A-D interconnect box. This may be changed in future versions of the NI software drivers. If you have two analog acquisition devices installed, the same is applied for each device: If only one channel is sending data on that device, the PFI7 to PFI0 jumper must be installed.

This appendix documents both of the 32-channel (16-bit) setups and the 64-channel setups of analog input available. A list of the location of channel numbers in the Analog Terminal Box for digital camera systems is given for the type of A-D setup. You may find it useful to make a copy of this chart ([Table B-2 on page B-12](#)) and use the column titled Setup Name to record the connections for your installation. The connections for a typical application using two AMTI Force Plates and ten EMG channels for digital cameras is shown for each setup.

Although connecting analog inputs is not particularly difficult, it is important that certain naming conventions be followed for the external data to work smoothly with supplementary Motion Analysis software such as **KinTrak** and **OrthoTrak**. This is described in [“Analog Signal Naming Conventions” on page B-4](#).

Performance Specifications

Cortex is capable of collecting up to 192 channels of analog data at any frequency between 60 and 5000 Hz. In newer systems, using the NIDAQ MX 8.0 or later software, analog rates can be much higher. The maximum rate can be up to 255 times the video capture rate, but performance may vary with different computers. The master digital camera provides the clocking signals to the A-D card in the A-D computer, which provides the phase-locking mechanism. You must connect the A-D cable from the master camera to the A-D interconnect box. Data can be collected in the pause mode or the run (live) mode, without any delay or drift between analog and video signals.

Installing NIDAQ Software on an Cortex Computer

For **Cortex 1.0** and later software, we recommend installing NIDAQ version 8.0 or later as it gives better analog performance and allows higher analog sample rates. With NIDAQ 7.1 through 7.4, the maximum analog sample rate is 5000 samples/sec for all channels. With NIDAQ 8 and above, you can go to higher rates (typically 20,000 samples per second) for all channels.

Note: NIDAQ version 7.5 DOES NOT WORK with any version of **Cortex**. Versions of NIDAQ software are available for downloading from www.ni.com.

If NIDAQ Software is Already Installed

1. Shutdown/power OFF the computer and remove the A-D unit.
2. Power ON the computer, go to **START/Settings/Control Panel, Add/Remove Programs**, select **NI_DAQ** and then select **Remove**.
3. When complete, shutdown/power OFF the computer, wait 10 seconds, power ON the computer, let the system boot up, and then log-in when prompted. It is necessary to have the computer boot without the A-D card or software. Proceed to the next step.

New Installation of NIDAQ Software

1. Install the **NI-DAQ software, version 8.0 or later**. Install all of the default entities that are checked, then shutdown/power OFF the computer. Note that **Cortex** users can use either NIDAQ 7.1 through 7.4 or NIDAQ 8.0 or later. **Cortex** uses the newer NIDAQ MX libraries whereas earlier versions of **EVaRT** use the "Traditional" (Legacy) NIDAQ libraries. Also, note that NIDAQ 7.5 does not work at all with **Cortex** applications.
2. Power ON the computer and let it boot completely without the A-D card installed. This will complete the National Instruments software installation. Once the computer is completely booted, Shutdown/power OFF the computer once again.
3. Install the A-D unit and power ON the computer. It should come up with the Hardware Wizard and the "Found new hardware" pop-up window. At this point, the computer will automatically install the NI-DAQ drivers correctly.
4. Shutdown/power OFF the computer one last time and then power it back ON.
5. Go to the National Instruments Test and Measurements software and select Traditional NIDAQ devices and then right click and select test panel.
6. Run through some of the channels to verify that the board is seeing the data. For example, have someone step on the forceplate.
7. Close and launch **Cortex** and connect to cameras. You should see that all cameras are found as well as the type of A/D unit installed.

Note: If you have NIDAQ 7.0 + drivers already installed onto your system, it is not necessary to un-install the software when upgrading.

Analog Signal Naming Conventions

When connecting force plates and EMG equipment to the **Cortex** system, certain requirements must be met and conventions followed.

Typically, forceplates are connected to the first channels of the A/D system and then the EMG channels. Specific Analog signal names for the forceplates must be used if **KinTrak** and **OrthoTrak** are used. These names depend on the forceplate manufacturer.

Kistler Forceplates

8 channels per plate:

- Analog Channel 1 connects to the F1X1 signal.
- Analog Channel 2 connects to the F1X3 signal, etc.

If there are two plates:

- Analog Channel 9 connects to the F2X1 signal.
- Analog Channel 16 connects to the F2Z4 signal.

The reserved names for **OrthoTrak** and **KinTrak** are:

PLATE #1: F1X1 F1X3 F1Y1 F1Y2 F1Z1 F1Z2 F1Z3 F1Z4
PLATE #2: F2X1 F2X3 F2Y1 F2Y2 F2Z1 F2Z2 F2Z3 F2Z4

Note: These naming conventions are already set up in the Analog panel.

For AMTI or Bertec Forceplates

6 channels per plate:

- Channel 1 connects to the F1X signal.
- Channel 2 connects to the F1Y signal.
- Channel 6 connects to the M1Z signal.

If there are two plates, it connects to Channels 7 through 12. The reserved names for **OrthoTrak** and **KinTrak** are:

PLATE #1: F1X F1Y F1Z M1X M1Y M1Z
PLATE #2: F2X F2Y F2Z M2X M2Y M2Z

EMG Signal Name Conventions

For **KinTrak**, you must specify the channel names in the **KinTrak** project definition as well as in the **Cortex** analog setup screen, which is saved in the **Cortex** project file.

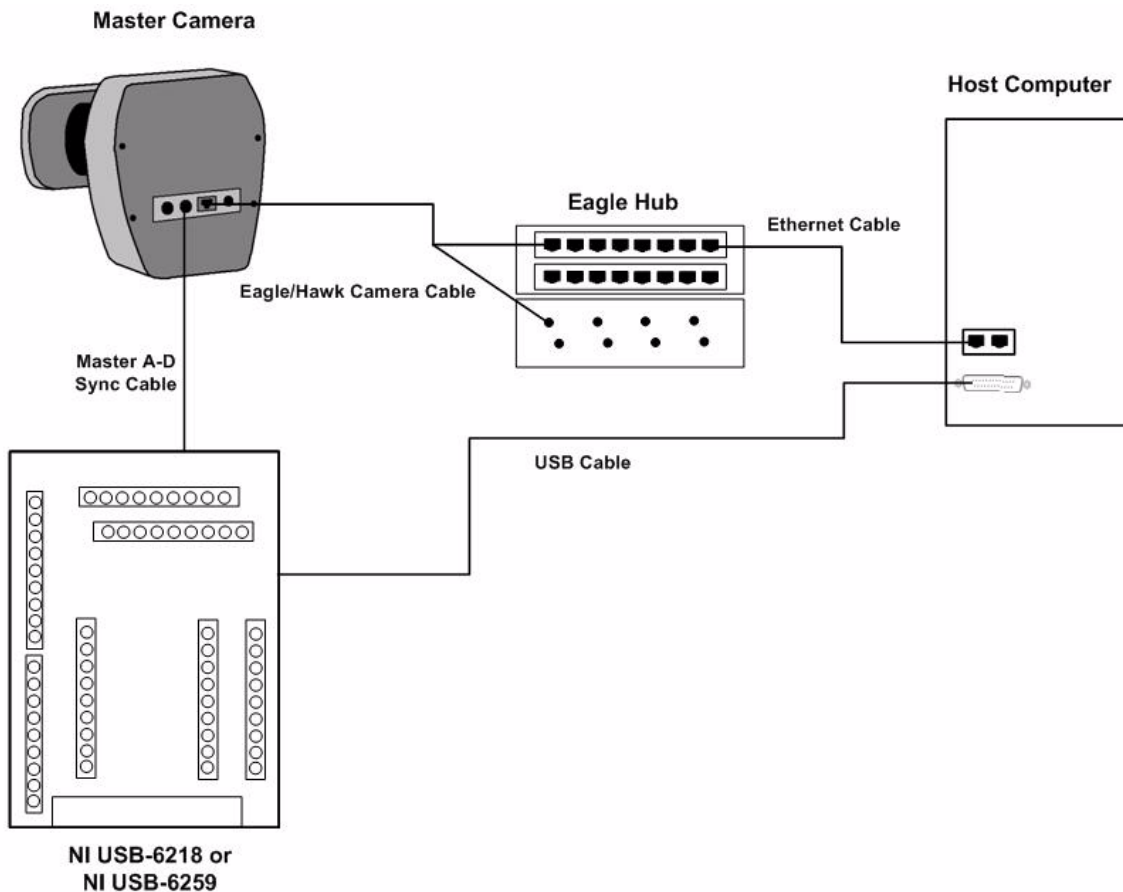
See the **OrthoTrak Reference Manual** for **OrthoTrak** muscle name conventions. The OrthoTrak muscle names are built into the analog display. They can be selected in the same manner as with the force platform manufacturer and channel.

32-Channel, 16-Bit NI USB-6218 Configuration

Note: The following has been tested on Windows XP Pro and Windows Vista with the latest web-based updates from Microsoft. The **Cortex** software will support up to 6 USB devices and up to 192 channels of analog input. USB 2.0 ports work best. USB 1.0 ports will also work but with reduced data rates.

Note: Up to 2 A/D devices can be installed into the host computer. The Sync Cable must be connected to both devices.

Figure B-1. 32-Channel, 16-Bit A-D Hardware Setup for the USB-6218 A-D



First: Install the Software

Before you plug the NI USB device into the host computer, you must first install the NI Acquisition software. For using the NI-USB 6218 A-D device, you need to have the **NIDAQ software version 8.3** (or later) installed. Run the National Instruments software installation, accept all the defaults, then let it finish and re-boot your computer. This may take 10-15 minutes. The remainder of the process takes less time. For more information, please refer back to [“Installing NIDAQ Software on an Cortex Computer”](#) on page B-3.

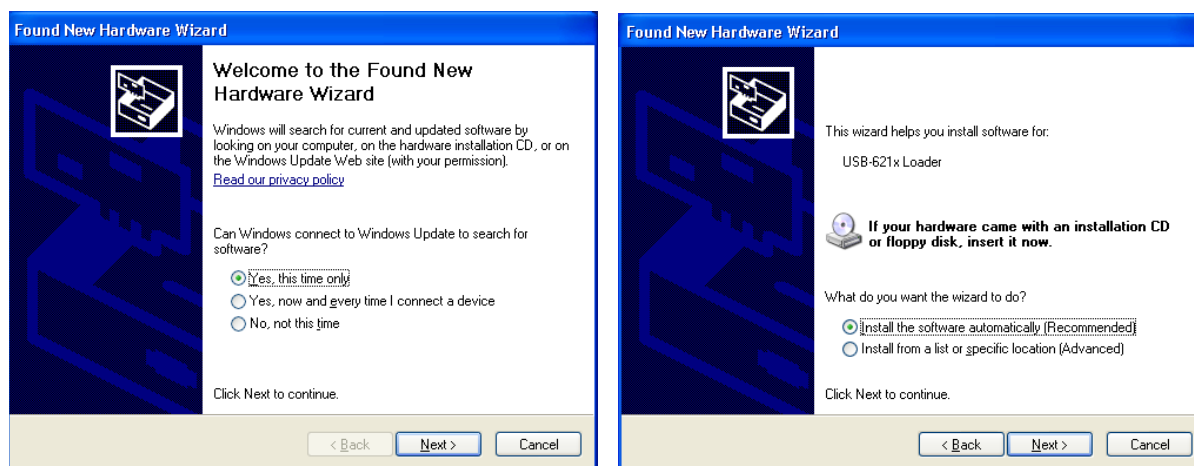
Second: Install the Hardware

1. Plug in the A-D device (NI USB-6218 will be used as the example below).
2. You will then automatically go through the “Found New Hardware Wizard” operation for a USB-device.

Note: You will do this two times. The first round is for the 621x Loader. The next round is then for the 6218 Device.

3. Select **Yes, this time only**, then **Next >**.

Figure B-2. Found New Hardware Wizard Interface—First Round



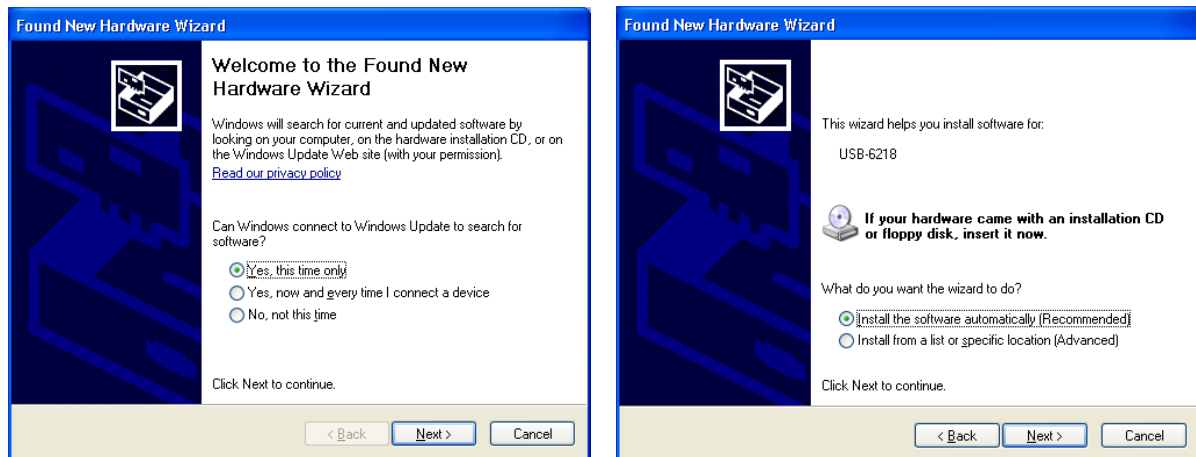
4. Click **Next >** then **Finish**.

Figure B-3. Completing the Found New Hardware Wizard Interface



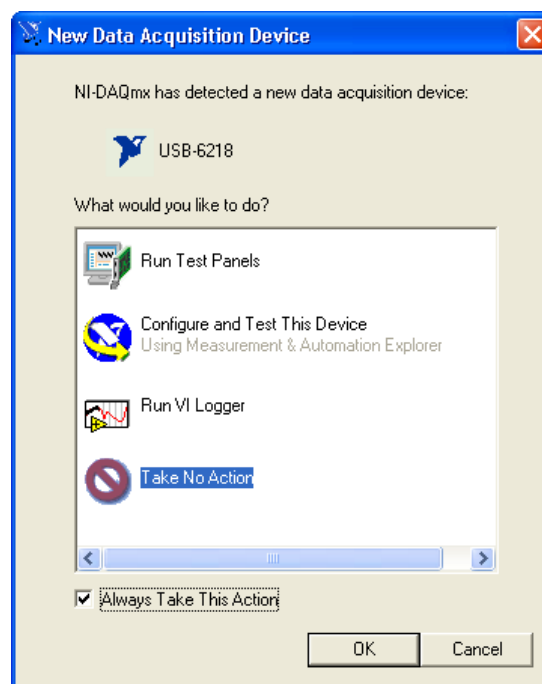
5. Now it repeats for the USB Device USB-6218.

Figure B-4. Found New Hardware Wizard Interface Repeated



6. Click **Next >**, **Next >**, then **Finish**. A message pops up that says the new hardware is installed and ready to use. Select **Take No Action**, and then check **Always Take This Action**.

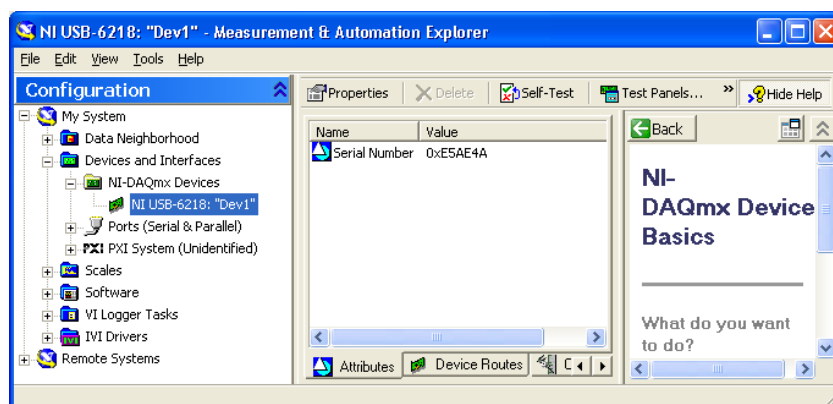
Figure B-5. New Data Acquisition Device Interface



Notes

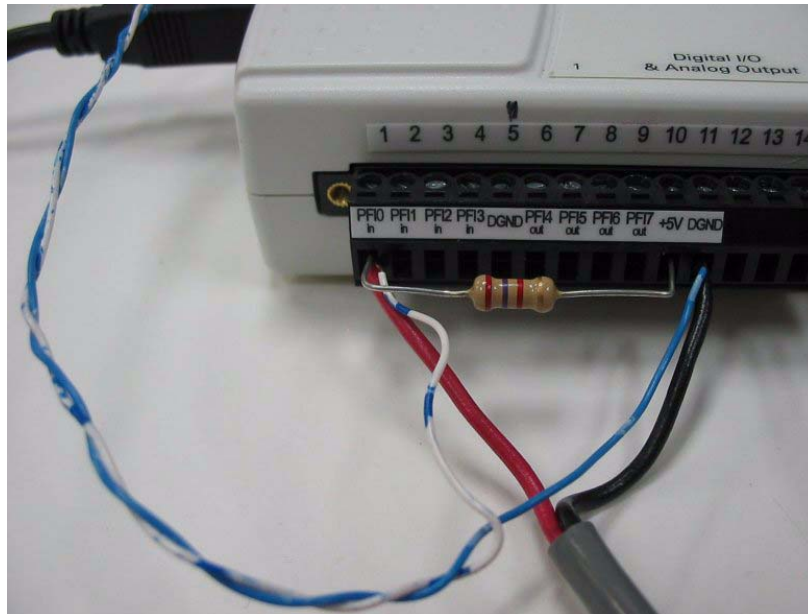
- You will get two message sequences for EACH USB A-D device that you plug in and you may get it again if you plug it into a different USB port, so it is simpler to plug into the same port each time.
- **Run Test Panels....** Setting the NI software to “Referenced Single Ended”: To get the correct looking signals on the Test Panels display, you need to set the **Analog Input > Input Configuration** to **RSE** (Referenced Single Ended). The **Cortex** software sets this mode as part of its analog setup procedures. Also, to have more than 8 channels displayed in the Test Panels (0-7), you must change this setting to **Referenced Single Ended** from its Differential Mode default.
- **Cortex** starts numbering the analog channels starting at 1 whereas the Test Panels and chart ([Table B-3 on page B-14](#)) starts numbers at channel 0.

Figure B-6. NI USB-6218 Configuration Interface



Installing the Clock Wiring from the Master Camera

Figure B-7. Clock Wiring on Side of NI USB-6218



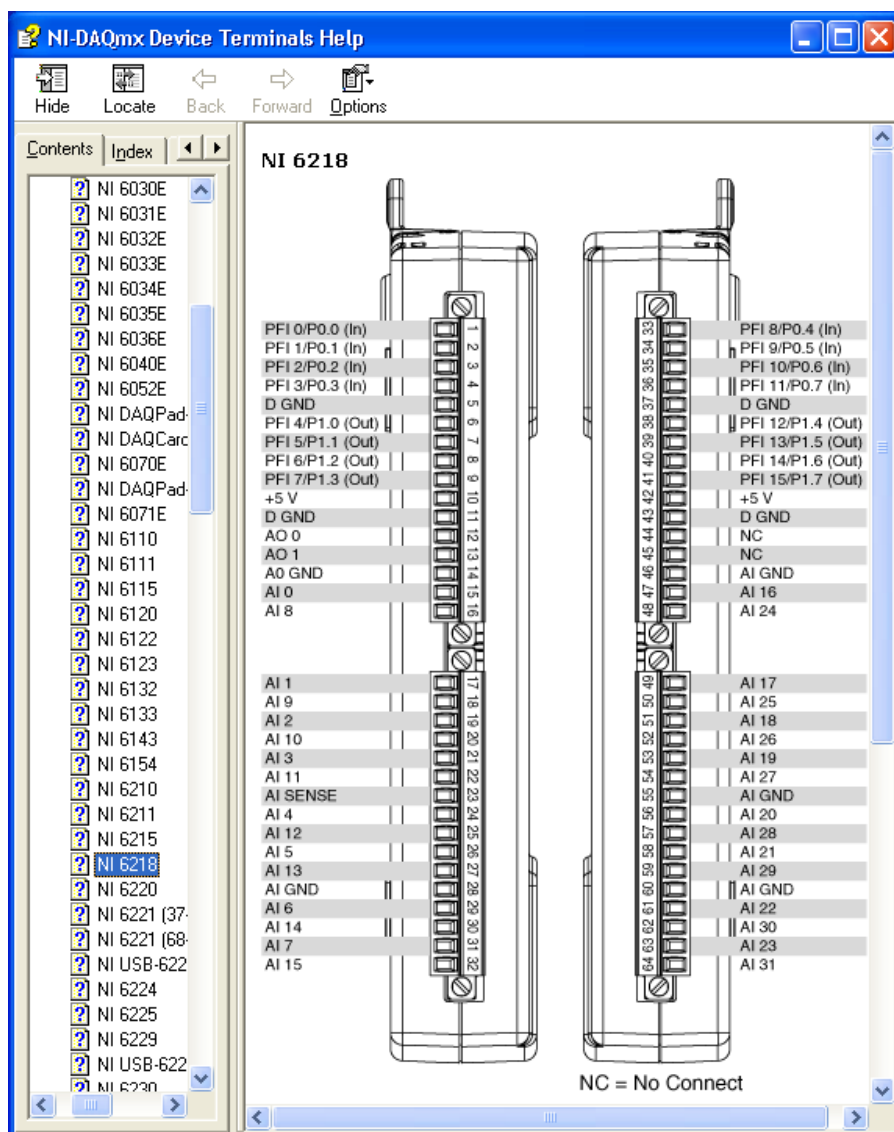
- Master Camera Red (or white): Connects to Pin 1 (PFI0)
- Master Camera Black: Connects to Pin 11 (D GND)
- 4.7 kΩ Resistor connects from Pin 1 (PFI0) to Pin 10 (+5 V)

Note: A jumper cable is required if multiple NI 6218 units are being used together (> 32 channels).

Note: NI USB-6259 uses PFI7 for clocking.

NI USB-6259 Analog Input Connections are listed in [Table B-3 on page B-14](#).

Figure B-8. NI USB-6218 Pinouts



The USB device pinouts are available online after you install the device. To find them:

1. Launch **Measurement & Automation** which was installed when you installed the NIDAQ software.
2. Select **Devices and Interfaces**, then **NI-DAQmx Devices** and the page should appear as shown above.

Note: NI Numbering starts at Channel 0 where **Cortex** channel numbering starts at 1. For example: AI 0 (above) corresponds to **Cortex** Channel 1.

Which USB Device is Channels 1-32, which is 33-64?

The first device you plug in should be channels 1-32 in your **Cortex** software. The second device should be channels 33-64 and so on for more devices. If you are not sure: When you connect to the Cameras in **Cortex**, the channel numbers (1-32) and the Serial Number of the USB device are reported in the dialog box. The Serial Number for the USB-6218 device is located on the bottom of the USB device.

Clock Wiring for Single and Multiple USB Devices

Eagle and Hawk cameras require a 4.7 k Ω pull-up resistor from +5 Volts to the PFI-0 pin as shown

Figure B-9. Clock Wiring for Multiple Devices



A Single Pull-up resistor with a 4.7 k Ω (or close) value will work for connecting up to 6 USB devices together as shown below. You need only to connect PFI0 and D GND signals in parallel for clocking the multiple USB Devices. A single pull-up resistor works for all devices.

USB 2.0 Cable Lengths

The USB-6218 comes with a one meter USB cable. It has been tested with longer USB cable lengths up to 16 ft. (5 meters) and it works well with any mix of cables.

USB Expansion Ports for Multiple A/D Units

Internal or External USB Expansion devices also have been tested and work well without problems. This allows you use a single USB port on the main capture computer and plug in as many USB devices as needed.

Table B-2. Analog Input Channel Connections and Master Camera Clocking (NI USB-6218)

Screw Terminal #	Cortex Channel #	Setup Name	Screw Terminal #	Cortex Channel #	Setup Name
1	PFI 0	4.7 k Ω Resistor & A/D Sync Cable & Jumper Cable*	33	PFI 8	Not Used
2	PFI 1	Not Used	34	PFI 9	Not Used
3	PFI 2	Not Used	35	PFI 10	Not Used
4	PFI 3	Not Used	36	PFI 11	Not Used
5	D GND	Not Used	37	D GND	Not Used
6	PFI 4	Not Used	38	PFI 12	Not Used
7	PFI 5	Not Used	39	PFI 13	Not Used
8	PFI 6	Not Used	40	PFI 14	Not Used
9	PFI 7	Not Used	41	PFI 15	Not Used
10	+5 V	4.7 k Ω Resistor	42	+5 V	Not Used
11	D GND	A/D Sync Ground & Jumper Cable*	43	D GND	Not Used
12	AO 0	Not Used	44	NC	Not Used
13	AO 1	Not Used	45	NC	Not Used
14	AO GND	Not Used	46	AI GND	Not Used
15	CH 1		47	CH 17	
16	CH 9		48	CH 25	
17	CH 2		49	CH 18	
18	CH 10		50	CH 26	
19	CH 3		51	CH 19	
20	CH 11		52	CH 27	
21	CH 4		53	CH 20	
22	CH 12		54	CH 28	
23	AI SENSE	Not Used	55	AI GND	Not Used
24	CH 5		56	CH 21	
25	CH 13		57	CH 29	
26	CH 6		58	CH 22	
27	CH 14		59	CH 30	
28	AI GND	Not Used	60	AI GND	Not Used
29	CH 7		61	CH 23	
30	CH 15		62	CH 31	
31	CH 8		63	CH 24	
32	CH 16		64	CH 32	

Note:* A Jumper Cable is required if multiple NI USB-6218 units are being used together (> 32 channels).

Note: NI Numbering starts at Channel 0 where **Cortex** channel numbering starts at 1. For example: AI 0 corresponds to **Cortex** Channel 1.

Maximum Analog Acquisition Rate

When measuring 64 channels with 16-bit resolution, the maximum analog rate is determined by one of the following considerations:

1. The maximum throughput of the National Instruments (NI) A-D product
2. The video sample rate multiplied by 255

The NI USB-6218 A-D unit, which is often supplied with Motion Analysis systems, has a maximum throughput of 250,000 samples/sec (per unit). So two units with 64 channels would have the ability to collect up to 500,000 samples/sec. Either way, it is 250,000/32 channels or about 7500 samples/sec per channel with all channels collecting data. If you cut down to 16 channels, you can have 15,000 samples/sec per channel. You can also connect up to 6 devices and use only 16 channels from each device.

For increased speed, use the faster USB-6259 A-D unit from NI. This unit is 32 channels, 16 bits (same as USB-6218), but has a throughput of 1.25 Million samples/sec or 5-times greater than the throughput of the USB-6218.

The video sample rate multiplied by 255 limitation comes into play mainly if a slow mocap video rate (e.g. 60 Hz) is being used. The max analog rate in this case is 15,300 (60 X 255). Under normal conditions, customers using higher analog rates would also be using higher mocap video rates. With a mocap video rate of 200 Hz, the max analog setting is 200 multiplied by 255 or 51,000 Hz. Note that you can do this with the USB-6218 with fewer channels turned on: $250,000 / 51,000 = 4.9$ (or 4 channels).

NI USB-6259 Analog Channel Connections

The following table ([Table B-3](#)) provides the analog channel connections and master camera clocking for the NI USB-6259 as interfaced in the **Cortex** software.

Table B-3. Analog Input Channel Connections and Master Camera Clocking (NI USB-6259)

Screw Terminal #	Cortex Channel #	Setup Name	Screw Terminal #	Cortex Channel #	Setup Name
1	CH 1		49	CH 21	
2	CH 9		50	CH 29	
3	AI GND		51	AI GND	
4	CH 2		52	CH 22	
5	CH 10		53	CH 30	
6	AI GND		54	AI GND	
7	CH 3		55	CH 23	
8	CH 11		56	CH 31	
9	AI GND		57	AI GND	
10	CH 4		58	CH 24	
11	CH 12		59	CH 32	
12	AI GND		60	AI GND	
13	NOT USED		61	NOT USED	
14	AI GROUND		62	AI GND	
15	NOT USED		63	NOT USED	
16	NOT USED		64	NOT USED	
17	CH 5		65	NOT USED	
18	CH 13		66	NOT USED	
19	AI GND		67	NOT USED	
20	CH 6		68	NOT USED	
21	CH 14		69	NOT USED	
22	AI GND		70	NOT USED	
23	CH 7		71	NOT USED	
24	CH 15		72	NOT USED	
25	AI GND		73	PFI 0	
26	CH 8		74	PFI 1	
27	CH 16		75	PFI 2	
28	NOT USED		76	PFI 3	
29	NOT USED		77	PFI 4	
30	AI GND		78	PFI 5	
31	NOT USED		79	PFI 6	
32	NOT USED		80	PFI 7	4.7 kΩ Resistor & A/D Sync Cable & Jumper Cable*
33	CH 17		81	PFI 8	
34	CH 25		82	D GND	
35	AI GND		83	PFI 9	
36	CH 18		84	D GND	
37	CH 26		85	PFI 10	
38	AI GND		86	D GND	
39	CH 19		87	PFI 11	
40	CH 27		88	D GND	
41	AI GND		89	PFI 12	
42	CH 20		90	D GND	A/D Sync Ground & Jumper Cable*
43	CH 28		91	PFI 13	
44	AI GND		92	D GND	
45	NOT USED		93	PFI 14	
46	AI GND		94	D GND	
47	NOT USED		95	PFI 15	
48	AI GND		96	+5 V	4.7 kΩ Resistor

* A Jumper Cable is required if multiple NI USB 6259 units are being used together (> 64 channels)

Appendix C *Marker Sets*

Topic	Page
Overview	C-1
Animation	C-1
Biomechanics	C-3
Examples	C-4
Developing Optimum Markers Sets	C-5

Overview

When deciding how to place markers for **Cortex**, it is important to realize that asymmetry is used by the software to distinguish left from right on the subject. Therefore, thigh markers may not be placed symmetrically, left to right, and a single marker might be placed on one shoulder to distinguish left from right.

Also, asymmetry is used to distinguish 3 markers linked together in a triangle. Therefore, the hand and thumb marker should not be the same distance from the wrist marker and should be well separated.

Another limit is the actual number of markers used. For a very detailed skeleton, you may be tempted to use a large number of markers. However, since each marker requires computation time, there is a practical limit to the number of markers used before the speed of real-time tracking is impacted.

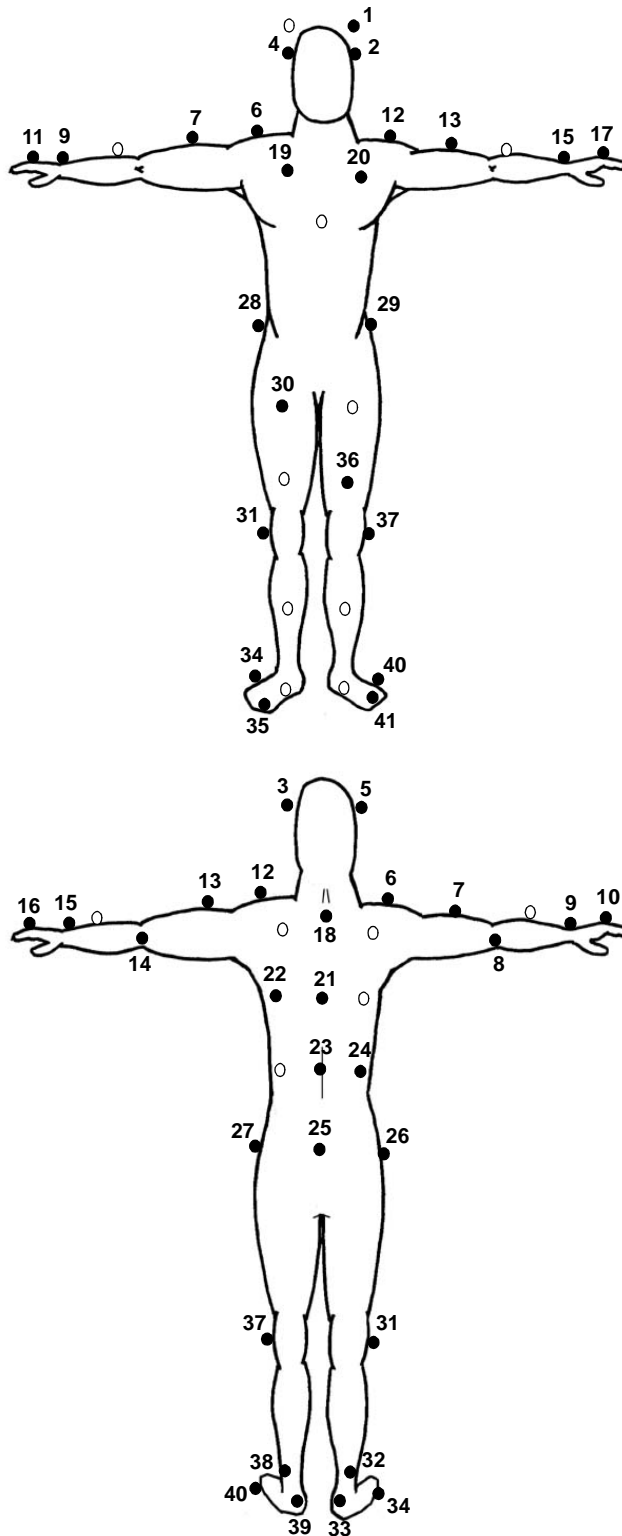
Specific examples of marker sets suited for both animation and biomechanics are given in the following figures.

Animation

[Figure C-1](#) is an example of a typical marker set using 35 markers. This example also shows suggested naming conventions. However, naming conventions that best suit your needs should be used.

Note: Biceps and thigh markers are intentionally placed asymmetrically to help the template distinguish left and right more easily.

Figure C-1. Typical Animation Marker Set



Note-When placing markers on end segments, the markers should not form a line and should not have mirror symmetry. Thus, thumb and hand markers should never be the same distance from the wrist marker and should be well separated.

- 1 - TopHead
- 2 - LFrontHead
- 3 - LRearHead
- 4 - RFrontHead
- 5 - RRearHead
- 6 - RShoulder
- 7 - RBicep
- 8 - RElbow
- 9 - RWrist
- 10 - RPinky
- 11 - RThumb
- 12 - LShoulder
- 13 - LBicep
- 14 - LElbow
- 15 - LWrist
- 16 - LPinky
- 17 - LThumb
- 18 - TopSpine
- 19 - RFrontShoulder
- 20 - LFrontShoulder
- 21 - MidBack
- 22 - LShoulderOffset
- 23 - LowBack
- 24 - RRootOffset
- 25 - Root
- 26 - RRearHip
- 27 - LRearHip
- 28 - RFrontHip
- 29 - LFrontHip
- 30 - RThigh
- 31 - RKnee
- 32 - RAnkle
- 33 - RHeel
- 34 - RMidFoot
- 35 - RToe
- 36 - LThigh
- 37 - LKnee
- 38 - LAnkle
- 39 - LHeel
- 40 - LMidFoot
- 41 - LToe

Biomechanics

When using **Cortex** in biomechanics applications such as **OrthoTrak**, the standard **Helen Hayes** marker set must be modified by adding one additional marker to either the left or right scapula. Also, new linkages must be added. This will give the asymmetry required so that the dynamic template can distinguish left from right.

In addition, the order of markers is important in real-time since the order in the list determines how quickly the software can establish marker identity using the dynamic template. In general, the marker list should start with the topmost marker. Proceed down the trunk of the figure, and then down each extremity.

For example, if head markers are used, they should be at the top of the list. If no head markers are used, the shoulder and pelvis markers should be at the top of the list.

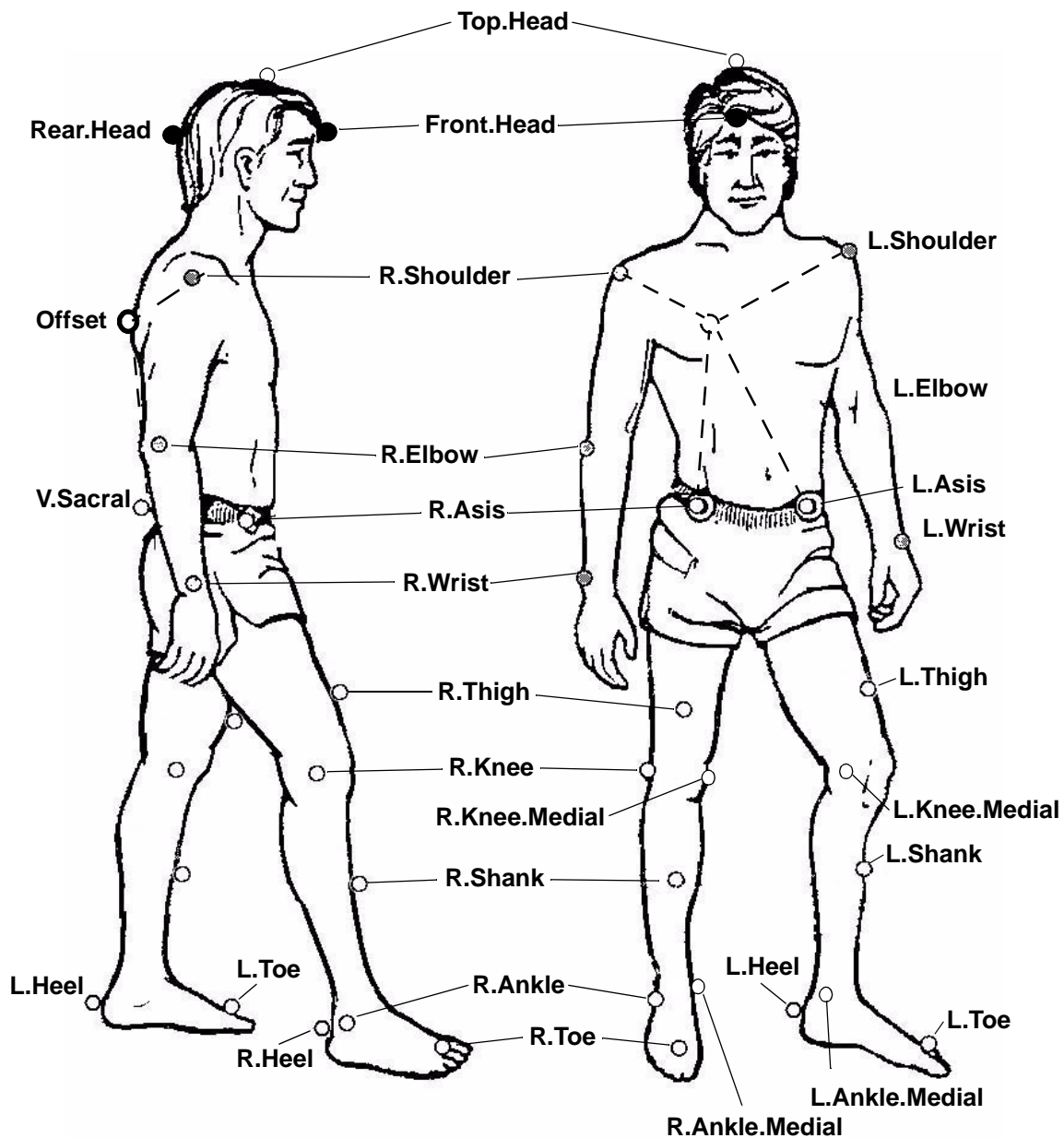
The recommended procedure is as follows:

Within Cortex

1. Launch **Cortex** and click on the **Connect** button.
2. Create a template with the modified marker set and save the results in the project file. For building a template, refer to [“Building a Template” on page 9-5](#).

You now have a template that can be used to automatically identify markers in real-time with this subject. When you click on the **Run** button and the subject enters the capture volume and all markers are visible, the linkages to the markers will appear automatically, indicating that the markers are properly identified.

Figure C-2. Helen Hayes Marker Set Marker Placement



Examples

Refer to the sample project folders in the **C:\ProgramFiles\MotionAnalysis\Cortex50\Samples** directory, which includes complete marker sets.

Developing Optimum Markers Sets

It has always been very tempting to anyone in the world of motion capture to get one “optimum” marker set. But typically, even the best animators and researchers use flexible marker setups, altering the marker sets to fit their desired capture goals.

If you are set on developing an optimal marker set, there are several things to take into consideration when you are trying to develop this.

1. What kind of movements are you doing?
 - Is there going to be a lot of bending at the waist? If so, then front hip markers are probably not good choice to use.
 - Are you going to be doing a lot of movements like rolling on the floor or sitting in a chair or laying down? If so, consider how you have markers on the back, since they will be obscured a lot.
 - Are you crouching a lot? If so, markers on the front of the body (chest) might be a bad idea.
2. Camera placement: Are you using a single camera placement scenario (that is will you move your cameras around)?
 - If yes, then you have more flexibility with a single, “optimal” marker set.
 - If no, then you have to take the issues listed in (1) above into consideration. Especially in regards to the movements where the markers are blocked by your subject's body.
3. Optimal number of markers: A general rule is that if you want a full 6-DOF set of information for each segment, you must have a minimum of 3 markers per segment. Currently we can shortcut that by allowing markers to be shared across joints (like the knees, ankles and elbows). Also, consider the Joint Virtual definitions, to get good quality Joint Virtual definitions, you want to have enough markers on the segment so that you can reconstruct missing markers using markers on that same segment. Typically in this case 4 markers per segment is advantageous. An example would be to place markers on the upper arm in the following positions: Shoulder, bicep, tricep and elbow.
4. Landmarking: Markers should be positioned, when possible, on bony landmarks. A bony landmark is an area like your ankle malleolus, elbow, knee condyles, wrist bones etc. This avoid the undo influence of soft tissue movements which can lead to noise in the marker positions.
5. Between people, the markers don't have to be in exactly in the same position. But a close approximation to the different sizes of people is required, especially if you are planning on using the PoseID-autofit option.

If you don't care about using the PoseID-autofit option, then you can place the markers anyway you like.

Appendix D *Capturing Facial Motion*

Topic	Page
Overview	D-1
System Configuration and Setup	D-1
Marker Placement	D-4
Building a Face Template	D-5
Examples	D-7
Facial Animation Techniques for Motion Capture	D-9

Overview

The **Cortex** system can also be used to capture the fine nuances of human facial motions. Three to six cameras, positioned up to 30 degrees apart around a relatively stationary subject, will provide sufficient coverage. The motion of 4 mm reflective markers, strategically placed about the subject's face, is captured and 3D translation data provides manipulation to an animated character's face model.

Animation programs like **SoftImage**, **Maya (Alias)**, **Motionbuilder**, and **3D Studio Max** will currently accept this data.

System Configuration and Setup

In order to use the **Cortex** system for facial motion capture, some additional equipment is required. This is known as the Facial Motion Capture Accessory Kit, and it contains a small 2D calibration L-frame, a facial marker set with glue, a mirror, and tweezers.

The longer focal length lenses allow positioning of the cameras an appropriate distance from the subject, resulting in ring light illumination that is evenly distributed across the field of view. The cameras should be positioned on a subject wearing the reflective markers.

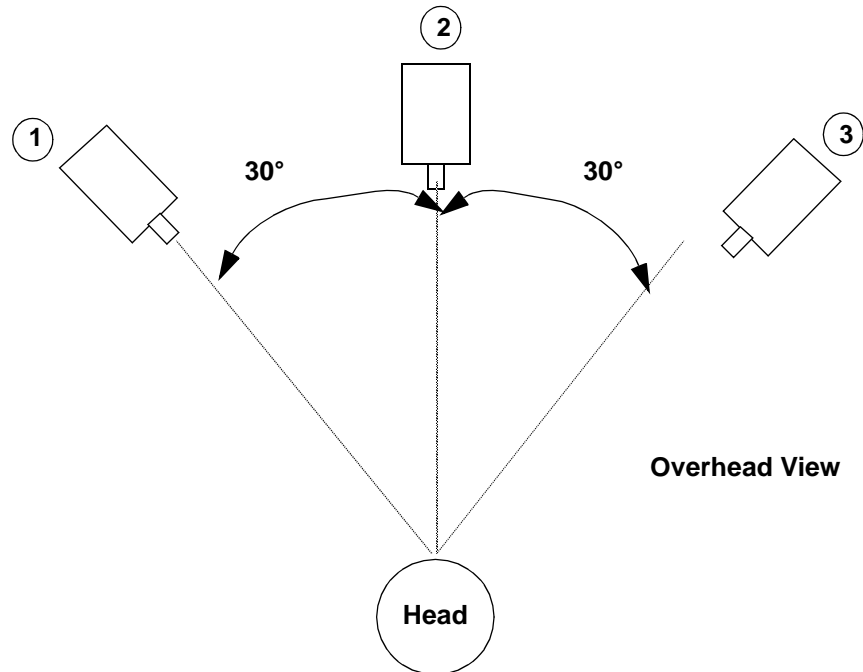
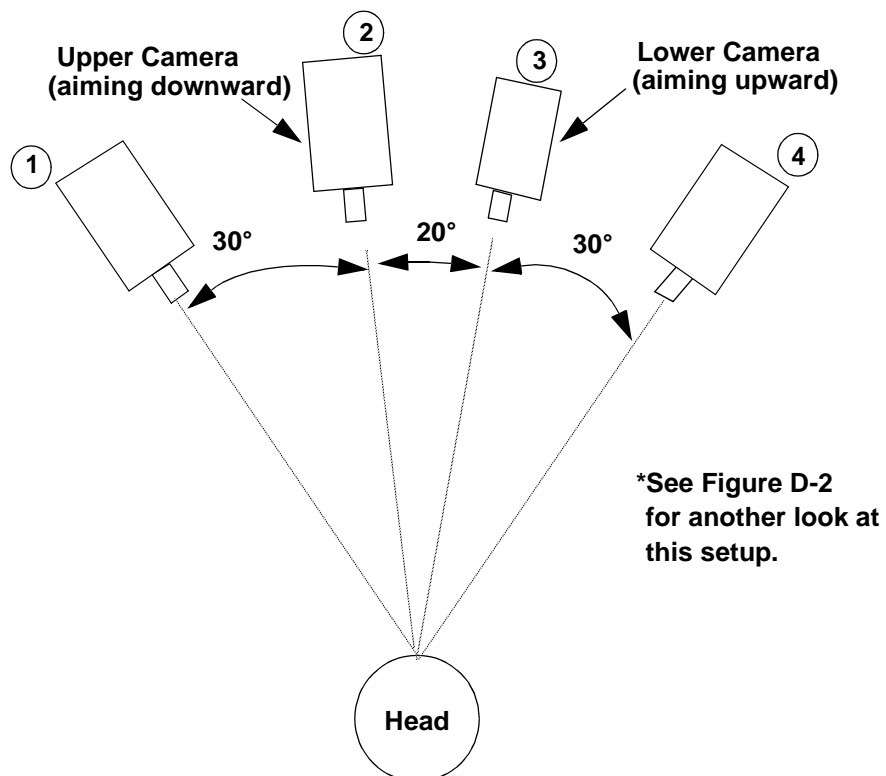
The goals of camera placement are:

- Have at least 3 cameras see as many markers as possible. When 3 or more cameras see a marker, the chance of a ghost marker occurring is minimal.
- Minimize the merging of markers and marker dropout in the camera views. Both are undesirable.

- Maximize and balance angular displacement between cameras by having at least 30 degrees of angular displacement between the cameras. The exception is the lower camera (see [Figure D-1](#)).
 - Optimize what each camera sees by ensuring that each field-of-view is filled with as many markers as possible.
1. Start by setting up the cameras as shown in [Figure D-2 on page D-4](#).
 2. Have the subject sit comfortably on a stool or chair facing the camera array.
 3. Optimize the subject-to-camera distance by ensuring that markers fill the field of view, but are not outside the field of view.
 4. Have the subject open their mouth wide, and make sure the head and chin markers stay in view. Look for potential merging between markers. This can especially be a problem around the lips.
 5. With the camera positions optimized, place tape on the floor marking where the legs of the stool or chair are located.
 6. Attach the facial calibration L-frame to the light stand and position it next to the subject. With the subject still sitting, adjust the height of the light stand until the square is positioned at the same height as the subject's head.
 7. Remove the subject and stool from the capture zone and position the calibration L-frame within this zone. The square is now located where the subject's head was located.
 8. Adjust the square's position so each camera sees as many calibration markers as possible. Remember that the subjects's face will be within this calibrated space during motion capture. Mark the floor with tape where the feet of the light stand are positioned. This will facilitate quick recalibration if it becomes necessary.
 9. As an alternative, you may put the facial calibration L-frame against a wall, calibrate, and then capture as long as the subjects face is within 1 foot of the facial calibration L-frame.

Figure D-1. Three and Four Camera Facial Motion Capture Setup

Note—This setup is given as a minimum for the required setup.

***4-Camera Facial Motion Capture Setup**

Marker Placement

The number and placement of markers for facial motion capture is dependent on the animation character's face model, and the animation software used to apply motion to the model. If a human face is to be animated, the markers should be placed at the major motion points on the face. If the face of a non-human character is to be animated, markers will be placed where the facial characteristics unique to that character will be accentuated.

In most cases, general areas of the face will need to be marked and captured. The following are suggested marker placements for facial motion capture:

Head

Three markers are used to identify head movements. If possible, the markers should be placed on areas with little or no skin movement. A tight fitting skull-cap may be used for attaching markers to the head. One marker should be placed on top of the head and one on each side of the head. These three markers are used to calculate the center of the head, which is the point from where all other marker translations are calculated.

Figure D-2. Marker Set for Facial Motion Capture



Eyebrows

One to three markers per eyebrow are used to track eyebrow movements. The exact position of markers on or around the eyebrows depends upon the subject's face.

Nose Bridge	Place one marker between the eyes, on the upper bridge of the nose. This area tends to be a junction point between the different regions of the face.
Eyelids	Both the top and bottom eyelids may be marked; however, you can expect some optical interference from the eyelashes, which can add more time to tracking and editing. Also, if the bottom eyelids are marked, these markers should be offset from the position of top eyelid markers to minimize marker merging.
Nose	The nose has relatively little motion except for the nostril. If nostril flaring is of interest, attach a marker to each nostril.
Cheeks	At least one marker should be placed on each cheek. Exact location will depend on the animation character model and the facial features of the subject.
Lips	The lips usually have the greatest amount of movement on the face. From 4 to 9 markers can be used to capture lip movement. Markers on the top lip should be offset from markers on the bottom lip to minimize merging. Also, areas around the lips can be marked to provide motion transitions.
Chin	One to three markers can be attached to capture chin motion.
Jaw	Attach one or more markers along the jawbone for jaw motion. This is very important for lip syncing.

Building a Face Template

1. Choose **Motion Capture** from the Mode Buttons.
2. Choose **Output** from the panel buttons.
3. Check the Tracked Binary (TRB or TRC) check box on the Output panel.
4. Type a file name in the name box and press **Enter**.
5. Set the Duration (seconds) to **10**.
6. Collect motion data of the subject by having the person stand in the middle of the capture volume.
7. Click **Record** on the Output panel.
8. The subject must stay in an initial frozen position for three to five seconds.
9. After staying frozen in this initial position for up to five seconds, the person must move through a complete range of facial motion that exhibits the full extent of stretch that will be experienced during subsequent motion capture sessions. Exaggerated motion must be avoided and all markers should remain in full view. This step should not require more than fifteen seconds.
10. After fifteen seconds passes from the moment **Record** was clicked, the system will automatically stop collecting and tracking marker data.

At this point, a Tracked Binary (TRB or TRC) file has been generated in the current directory and is ready for editing. Next, the markers must be hand identified according to the marker list built for the subject's marker set.

11. Choose **Post Process** from the Mode Buttons.
12. Click **Quick ID** and identify the unnamed markers according to the conventions described in [Appendix C, Marker Sets](#).
13. Click **Rectify**.
14. Manually cleanup and identify all tracks in this range of the motion file.
15. Click **Create Template**.
16. Select **Face Template**.
17. Select the appropriate Frames Range:
 - **Current**—the current displayed frame
 - **Selected**—frames highlighted in blue, low to high in dashboard
 - **Visible**—what is displayed across the screen, as a function of the time zoom
 - **All Frames**—all frames
18. Click **Create Template**.

Note: You must use the Objects panel to select the marker sets.

Face Template Considerations

1. Face templates link all markers invisibly to other markers for the Template ID and Streaming ID functions. With all markers linked to other markers, the Template ID works much more quickly.
2. Explicit linkages in the Marker Set are used for the Streaming (Real Time) Rectify and Post-Processing Rectify along with the two Linkage Stretch parameters. Generally the face template with many linkages works better than one with fewer linkages for Rectify to work. Use only relatively rigid links and asymmetric markers if possible.
3. Keep the Linkage Stretch Parameters in the **Motion Capture > Tracking** panel at 7 and 5 (or close) to prevent mis-IDs and allow high enough performance.

Examples

A Face Close-Up Tutorial

For an example of facial motion capture data, refer to the **C:\Program Files\Motion Analysis\Cortex50\Samples\Dave Face Stabilization** directory and open up the project file **FaceOnly.prj**.

With this project, you can review how face marker data is tracked with body marker data. The **FaceOnly.prj** project file defines a 17 marker face capture template.

To play the motion capture data, load the VC file **DaveFaceCloseup1.vc**. This was a capture done with a close-up view of the face taken by the video camera.

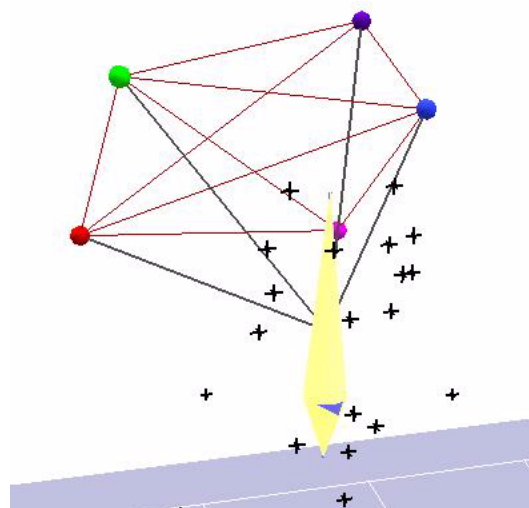
To see the video data:

1. Open another window and select a Data View type of "Full Color Video".
2. Right-click in the video window to bring up the AVI Frame Offset input dialog.
3. Set the value to -48.

This properly aligns the video data with the motion capture data.

This directory also contains a set of example files demonstrating how to use the marker stabilization tool in **Cortex**. The file **Head.prj** defines a marker set of just the head markers for the performer (these markers are separate from the face markers). These head markers were tracked and exported to a TRC file—**Head.trc**. This TRC file was then used with Calcium and a single segment skeleton was created with the only segment being called "Head". This creates a 6DOF segment which exactly tracks the motion of the head of the performer.

Figure 14-2. "Head" Segment and its Driving Markers

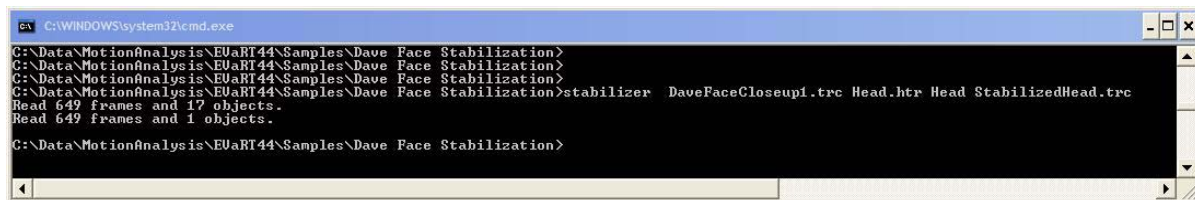


The project file is called **Head.prj** and it contains the Calcium setup information.

In **Cortex**, the **Head.prj** file is loaded and the skeleton generating tools are turned on. In addition, the selection of the Streaming Option **Make object A relative to segment named "Head" of Main object** is turned on.

For each TRC file that you capture, you calculate then export the HTR skeleton to an HTR file. The TRC file and HTR file are used by a stand-alone command line program called "Stabilizer" (from the Mocap Toolkit) to generate a stabilized TRC file from the original TRC file. The stabilizer command would look as shown in [Figure 14-3](#).

Figure 14-3. Stabilizer Command Dialog

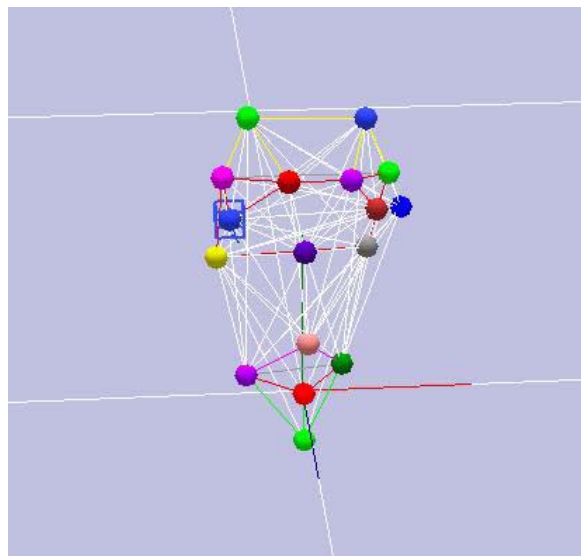


```
C:\WINDOWS\system32\cmd.exe
C:\Data\MotionAnalysis\EUaRT44\Samples\Dave Face Stabilization>
C:\Data\MotionAnalysis\EUaRT44\Samples\Dave Face Stabilization>
C:\Data\MotionAnalysis\EUaRT44\Samples\Dave Face Stabilization>
C:\Data\MotionAnalysis\EUaRT44\Samples\Dave Face Stabilization>stabilizer DaveFaceCloseup1.trc Head.htr Head StabilizedHead.trc
Read 649 frames and 17 objects.
Read 649 frames and 1 objects.
C:\Data\MotionAnalysis\EUaRT44\Samples\Dave Face Stabilization>
```

This indicates that the **DaveFaceCloseup1.trc** file is to be stabilized by the **Head.htr** file using the segment named "Head" and the output is to be **StabilizedHead.trc**.

The face markers will be repositioned such that the motion of the head segment is removed. This effectively places the face markers at the origin of the data space, as shown in [Figure 14-4](#).

Figure 14-4. Face Markers at Origin of the Data Space



You can view this TRC file by loading the **FaceOnly.prj** file into **Cortex** and loading the tracks file **StabilizedHead.trc**.

This data is now ready for use in a facial animation system.

Facial Animation Techniques for Motion Capture

Types of Facial Animation

The goal of any facial animation technique is to move the geometry of the face around in a meaningful way. The way the mesh is modified must look very convincing to the eye since people are very attuned to facial motion and any anomaly will be quickly picked up. The two basic types of mesh modification used for facial animation are morphing and direct mesh deformation.

Mesh Deformation

Mesh deformation is a direct manipulation of the facial mesh. Markers are placed on the mesh and connected, such that, as the marker moves around so does the mesh. Each marker is given an area of influence on the mesh (areas of influence may overlap) that fades away the farther away the mesh is from the marker. Within any particular animation system, this technique is often identically the same as what is used to do full body skinning. This is not to be confused with clustering, where groups of markers are lumped together under one control handle (the cluster). For example, as the handle moves around so do the markers as a single group. Clustering is frequently used in facial animation but usually as a way of creating faces used for morphing.

Figure D-3. Face Model in Base Position with a Set of Markers

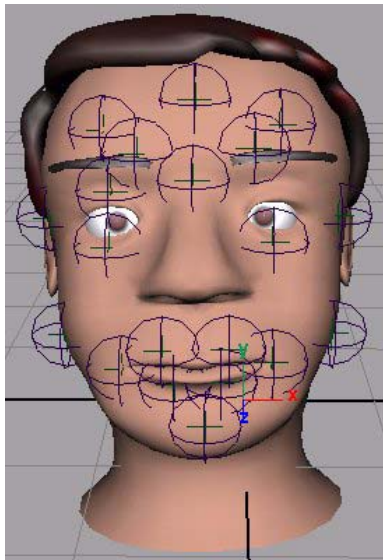
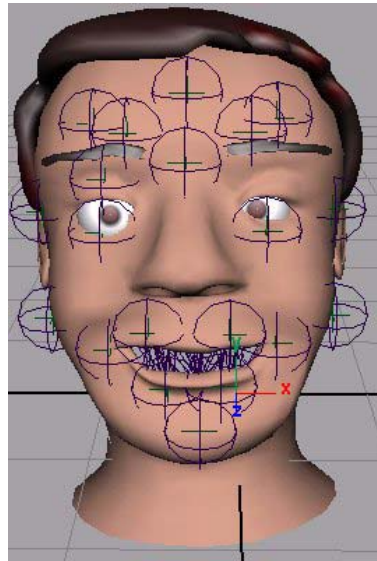
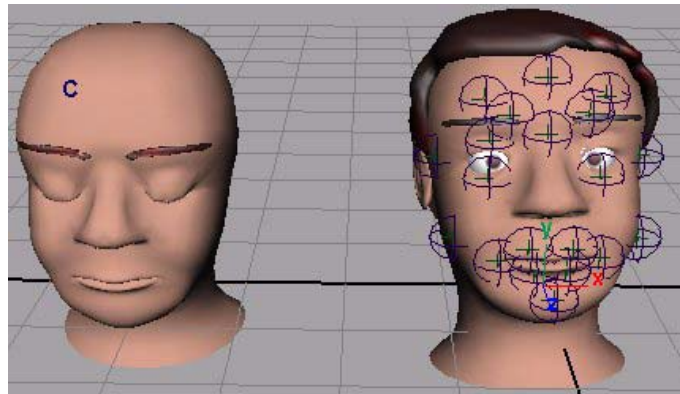


Figure D-4. Face with Motion Capture Data Applied to the Markers

As the markers move around they pull the mesh with it. Care has to be taken on the areas of influence, especially around the mouth, so that markers affect only what they should affect. The upper lip markers, for example should not influence the lower lip.

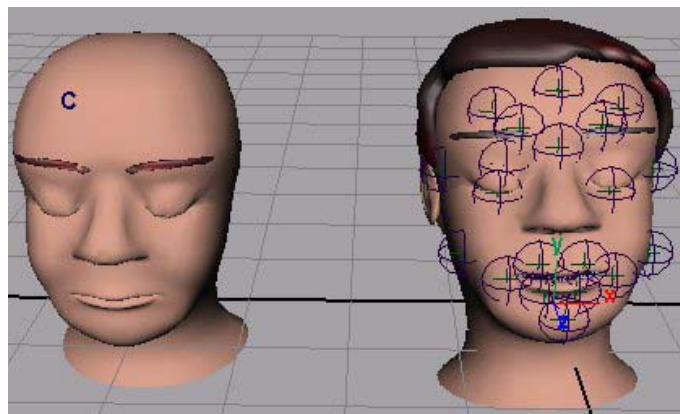
Morphing

Mesh morphing is by far the most commonly used facial animation technique. It is an extremely powerful and easy to use technique. Like with the mesh deformation technique an animator starts with a base face. Copies of the base face are made, each copy is modified into a different facial expression (open mouth, smile, eye blink, etc.). To create a blend of these expressions, the animator specifies how much of each expression is used to compose the result. For example, 50% open mouth, 10% smile, 100% eye blink. Note that the percentages do not have to add up to 100. The animator only specifies how much of each expression goes into the face. Often times the expressions are called sub-expressions, morph targets, or simply targets.

Figure D-5. Base Face with the Eye-blink Expression to the Left

This is the same face as used in the mesh deformation example. This demonstrates how it is possible in some animation systems to combine the techniques for even more powerful results.

A careful inspection of the base head image with the markers should reveal that there is only one eye-lid marker. It is on the right eye of the character. This particular facial motion capture data set had only one eye-lid marker so it would not be possible to use the mesh deformation technique to animate the eye-blinks of both eyes. Only the right eye could be used. However, using the morphing technique, the up-down motion of the right eye-lid marker can be used to control the contribution of the eye-blink morph target.

Figure D-6. Result of Combining Techniques

The right eye-lid marker has moved down a small amount to indicate that the actor blinked. Using the morph target on the left, both eyes appear closed when the target expression is applied to the final result on the right.

Unlike the mesh deformation technique, the morph target technique uses multiple meshes to do the facial animation. One requirement for all the morph target meshes is that they have exactly the same topology (the exact same number of vertices and polygons with exactly the same con-

nections between them). Therefore, it is the usual practice to model the base face first, then make copies of it for modifying into other expressions.

Using Motion Capture Data with Facial Animation

This section describes how motion capture data is used with each of the facial animation techniques. The direct mesh deformation technique is extremely well suited to using motion capture data while morphing is very badly suited to using motion capture data. By themselves, each technique has its advantages and disadvantages. The correct answer for an animation system lies in the ability to use both simultaneously which allows the animator to have the best of both worlds. Each is described in the following subsections.

Facial Retargeting with Offsets for Mesh Deformation

The marker placement on the actor rarely coincides exactly with a corresponding marker placement on the character. A character's face is almost always exaggerated in some fashion that makes it impossible to find an actor to exactly match it. However, without careful placement of the markers on the mesh of the character, the deformation of the mesh simply will not work. The solution which satisfies both of these problems is to use the motion of the actor as an offset from the base position of the character.

To use this solution, you need to create a marker set on the mesh of the character that has the same number of markers with the same names as the marker set of motion capture data from the animator. The only difference between the two sets of markers is their starting positions in the base (neutral) pose of the faces. Rather than using the absolute position of the markers from the actor, you calculate the offset of motion of a marker from its base position. That is, how far it has moved from its starting point. This offset is what you apply to the character's markers. This way it doesn't matter if the character's face is really wide or really long or otherwise oddly proportioned. The starting points of the markers will always make sense and their motion from the actor will almost always work.

You can even scale the magnitude of the offset motion to exaggerate the motion or to dampen it.

The advantage to this technique is that it is technically easy to understand and implement. The drawback is the lack of high level expression control for the animators. If there are expressions that couldn't be captured (or weren't captured) it is hard to use keyframe animation on the mesh deformation markers to create totally new expressions.

Gesture Recognition for Morph Targets

The advantages and disadvantages for using morph targets are somewhat reversed with respect to the mesh deformation technique. Morph targets are much easier to use by animators to control expressions at a high level. Standard keyframe animation techniques work well with morph targets. On the other hand it is extremely difficult to create a general purpose ability to use motion capture data within such a system. Some simple and useful exceptions are found for parts of the face (such as the eye-lid example used above, or perhaps the jaw) but for some parts of the face, most notably around the mouth, it is very hard.

Most facial expressions do not limit themselves to a single spot on the face (such as the location of a single marker). Each facial expression moves many markers at once. A smile, for example, not only moves all of the markers around the mouth but it also moves markers around the eyes, temples and forehead (most people squint when they smile). So there isn't an easy way of linking a smile morph target to one or two motion capture markers. A smile is a true smile if and only if a whole set of markers move in just the right way. Systems capable of doing this kind of analysis have been made for doing realtime facial animation but this technique has not yet found its way into most animation systems.

Currently, the most advanced facial animation systems (the ones used to make some of the popular feature length animated features) use knowledge of skeletal and muscular anatomy to understand how the underlying tissue affects the skin movement of a character. These complex animation systems are all morph targets based in the sense that the animator still works with a blend of high level expressions to achieve their final result. They still say "I want half a smile" and "part of a smirk". The animation engine accounts for muscle and bone movement while composing the final result for the skin.

Other Facial Animation Inputs

Motion capture data isn't the only kind of input used for doing facial animation. A summary of other techniques is given here.

Keyframe Animation

All animation systems have keyframe animation at their core. The ability to use keyframe animation in conjunction with motion capture data is vital to getting the best overall results. It is important to use the keyframe tools without damaging the motion capture data. Some keyframe animation systems require that the motion capture data be simplified in order to control the data with keyframes. This is a mistake of large proportions as motion capture data should never be decimated. Motion capture data can't quite get everything that an animator will need from the motion so extra motion has to be layered on top. This is true of all motion capture types (face, body, hands) and devices.

Phoneme Recognition

Lip-synching is an important sub-problem of facial animation. The motion of the face (particularly the lips) must be synchronized to the audio track of the voice talent. The classic technique is for the animator to listen to the audio track and keyframe animate the facial expressions to match. The first part is to get the mouth in the right position to match the syllable. If the animator is using a morph target technique, it is very common to have a series of morph targets, each representing a common phoneme used in speech. This makes it straightforward, but time consuming to animate.

There are a number of automatic phoneme recognition technologies available for evaluating audio input and generating phonemes. This information can be used as a source of animation data for facial animation. Not only is phoneme recognition hard to get right, but the general approach has inherent limitations. Phoneme recognition can be helpful, but will never entirely solve the problem of facial animation. A summary of the drawbacks is as follows:

- **Too precise**—At any given moment the phoneme generator gives you only one phoneme (this will improve in the future). There is no notion of blending between them. All people slur their syllables to one degree or another, this kind of information is missed.
- **Anticipation**—Almost all sounds require a setup motion for the face. You open your mouth before you actually say anything, the audio track doesn't have that kind of information.
- **Other facial movement**—When people talk, their necks, ears, and other parts of the face move around significantly. This information is not conveyed. Lack of eye blinking information is perhaps the most important.
- **Non-audible facial expressions**—Most people intersperse their conversations with a variety of facial expressions to punctuate the conversation.

Despite these limitations an automatic phoneme generator can provide an excellent first pass at facial animation with the intent of going back over it to augment it with more facial animation information.

Eye Movement

Eye motion is a subtle, but a vital part of facial animation that must be present. A variety of techniques exist for obtaining eye motion, but it is difficult to get without hampering the acting talent. The most common approach is to take a video image of the eye and track the eye movement from the video footage. Some techniques track the whites of the eyes, others the pupils. Some use visible light, others use infrared.

In all cases, a 2D image is used to generate information about the translational movement of the eye in the image. This is then turned into rotation information to rotate the eyeball of the character.

Waldos

Waldos are physical input devices use as puppeteering controls for characters. Each input type is given a high level meaning such as head rotation, eye-blink, or a particular facial expression. For this reason, waldos and morph target facial animation systems work well together.

Other Motion Capture Issues

A miscellaneous collection of issues which affect, or are related to, facial motion capture are detailed in the following sub-sections.

Marker Size and Capture Volumes

Marker size, camera resolution, and capture volume calibration are all key elements to determining how large the capture volume can be and how much of the performer can be captured at once. Camera refresh rates and the number of markers can affect this too, although they're not as important. The latest camera systems have higher resolutions and the latest software has easy to use techniques for handling lens distortion and capture volume calibration. This adds up to the ability to have smaller markers and more of them which allows for full body and face motion capture simultaneously.

Having face and body data at the same time is an important technological hurdle. It simplifies a great number of face-body coordination issues and allows for real-time processing of the data so that live motion capture sets can be created (a live set is when the data is acquired, processed, applied to a virtual character and rendered at 30 frames a second).

Marker Stabilization

Sometimes the facial animation techniques require working with the data in a simple reference frame as though the head were an object by itself sitting on a table top. If the facial motion capture data is captured as part of a full body and face capture, then the facial markers have to be segregated and recalculated relative to the motion of the head segment of the body. This process is known as stabilization. This is a vital tool for keeping the facial data under control.

It is particularly important to have good, solid head motion in your character skeleton if you use this technique. Any amount of slippage in the motion of the head relative to the facial markers will result in jittery, noisy face data.

Sync to Body Capture

Ideally, the face and body are captured simultaneously so that the data is automatically synchronized because it's all part of the same data set. The good news is the latest motion capture systems allow for this. Global time information needs to be encoded in all motion files so that they can be later synchronized if they're not captured simultaneously. Time stamp information is always useful to have in any case.

Motion Capture of Hands

The motion capture of hand motion, and the fingers to be more specific, presents many of the same issues as facial animation. The capture volume limitations are about the same since the markers are about the same size. It is even common to use morphing techniques on the hands like on the face rather than treating the hand as a mini-skeleton and animating it like you would animate the full body. Certain hand gestures are very common so it's effective to model those few (fist, flat hand, pointing a finger) and morph between them. If motion capture data is used, however, skeleton animation techniques are easier to use and apply to the hands.

Limitations

Facial motion capture does have some limitations, not all the information that an animator might want can be captured directly from the face of the actor. Some examples are:

- **Tongue**—You can't put markers on the tongue. There is no effective way of capturing the full motion of the tongue with any kind of technique.
- **Neck**—Various regions of the neck move while talking. The motion of the tongue causes the underside of the jaw to move while swallowing and breathing causes other areas to move. Extra markers could be placed under the jaw and around the neck but then visibility issues become a concern.
- **Eyes**—Markers can't be used to track the eyes. Other techniques might be used for this.
- **Curl and other twisting motions**—The skin of the face doesn't just travel in straight lines. Many parts of the face scrunch and curl around. Pursing one's lips or pouting motions cause the lips to bend around in a variety of ways. Markers do not directly transmit this information, only careful placement of groups of markers can effectively sense this.
- **Number of markers**—Ideally it would be best to use as many markers as you can put on the face. Capture limitations prevent this so much information that might theoretically be measured will have to wait for higher resolution cameras and even smaller markers.

Despite this list, facial motion capture is, by far, the best overall technique for generating facial animation from a performer. No other kind of system is as versatile or productive.

Production Issues

Many practical issues creep into the animation process that do not have much to do with techniques for animation but rather with the process of animation itself. That is to say, it has to do with the relationship between the animator and the production tools. This is true for all tools, especially facial animation tools. Here is a list of issues that anyone doing facial animation should keep in mind:

- **Requirements do change**—Real productions do not march relentlessly from front to back, animators have to go back over the data many times to get the result they need. For example, a common mistake in morph target facial animation is the realization that you need more flexibility in a certain part of the mesh so you can make a new target expression that you suddenly discovered you needed. Since all the expression meshes must exactly match, you have to update all the existing meshes to incorporate your new change. This can be tedious and error prone. Some morph target systems do automatic updates for you. Be ready for this when setting up your workflow. Consider the possibility that you might, at times, have to work backwards through problems.
- **Facial animation is just one element**—Facial animation would be somewhat easier if all you had to worry about is the face. The problem is that you almost always have to attach the face to a head (and therefore to a body). The relationship between the face and body elements must be considered when setting up a character for animation.
- **Interactive versions versus full render versions**—The high resolution final images (if that is the final output) require high resolution facial meshes. This sometimes hampers the inter activeness of the animation system. The ability to use low resolution meshes for interactive work and then replace them before the final product is a valuable production tool.
- **Synchronization**—All input data needs to be synchronized. Global time information needs to be present in all the data so that as the animator works with different kinds of data (face, body, voice, video), they can be matched up in time.
- **Output**—The final output of motion data is not always a final image. The final output might be animation data sent to a game engine (or some other kind of interactive environment). In which case, the algorithms which underlay the facial animation need to be present in the game engine so it can reproduce the facial animation interactively.

Appendix E *Forcepla.cal File Format*

Topic	Page
General Information	E-1
Forceplate File Data	E-3
Forceplate Scaling Factor, X-Width and Y-Length	E-5
Using AMTI and Bertec Forceplates	E-6
Using Kistler Forceplates	E-9
Using Kyowa Dengyo Forceplates	E-10

General Information

Up to eight forceplates may be placed within the video capture space to measure gait forces. While **Cortex** gathers video data, it simultaneously acquires forceplate data.

To accomplish this, the forceplate output is connected to an analog input card in the **Cortex** system. The forceplate data is interpreted using a file called **forcepla.cal**. When **Cortex** reads in the trial data, it first searches the current directory where the project resides for the **forcepla.cal** file. If none is found, it then searches in the directory:

C:\Program Files\Motion Analysis\Cortex5\Samples\Example Forcepla.cal Files.

The **forcepla.cal** file contains information describing the location, orientation, and calibration of each forceplate used. The exact form of the file will depend on the forceplate manufacturer. [Figure E-1](#) shows the file form for Bertec and AMTI forceplates. [Figure E-2](#) shows the file form for Kistler forceplates.

The **forcepla.cal** file contains no text, only numbers. For multiple forceplates, the data for each forceplate in the system is included in one **forcepla.cal** file (see [Figure E-4](#)).

Note: The **forcepla.cal** file must be in the same directory as either the **Cortex50.exe** file or the *.prj file. Otherwise, the forceplate outlines will not appear in the 3D collection view.

Note: **Forcepla.cal** files in the past have been named with a “t”, as forceplate. Be sure to check that there is no “t” in **forcepla.cal**.

Example **Forcepla.cal** files for each type of forceplate (AMTI, Bertec, and Kistler) can be found in the directory: **C:\Program Files\Motion Analysis\Cortex5\Samples\Example Forcepla.cal Files.**

Figure E-1. Forcepla.cal File Structure for Bertec and AMTI Forceplates





Line#	Description	
1	Forceplate number (1 through 8)	
2	Forceplate scaling factor and [optional length and width of forceplate] (25 for AMTI setup with amplifier gain switches set to 4000)	
3		6x6 forceplate calibration matrix (Inverted Sensitivity) provided by the manufacturer
4		
5		
6		
7		
8		
9	Xo Yo Zo True XYZ origin relative to the geometric center of the forceplate—in cm (provided by the manufacturer).	
10	Xc Yc Zc XYZ location of the geometric center of the plate with respect to your video coordinate system. (the video calibration system's origin)—measured in cm	
11		3x3 forceplate orientation matrix to make the forceplate coordinate system match the laboratory coordinate system
12		
13		

Figure E-2. Forcepla.cal File Structure For Kistler Forceplates.

Line#	Description	
1	Forceplate number followed by "K" to indicate a Kistler forceplate.	
2	Forceplate scaling factor and [optional length and width of forceplate]	
3		8x8 forceplate calibration matrix created by the user
4		
5		
6		
7		
8		
9		
10		
11	Xo Yo Zo position of the forceplate transducers in cm (provided by the manufacturer)	
12	Xc Yc Zc XYZ location of the geometric center of the plate with respect to your video coordinate system. (the video calibration system's origin)—measured in cm	
13		3x3 forceplate orientation matrix to make the forceplate coordinate system match the laboratory coordinate system
14		
15		

Forceplate File Data

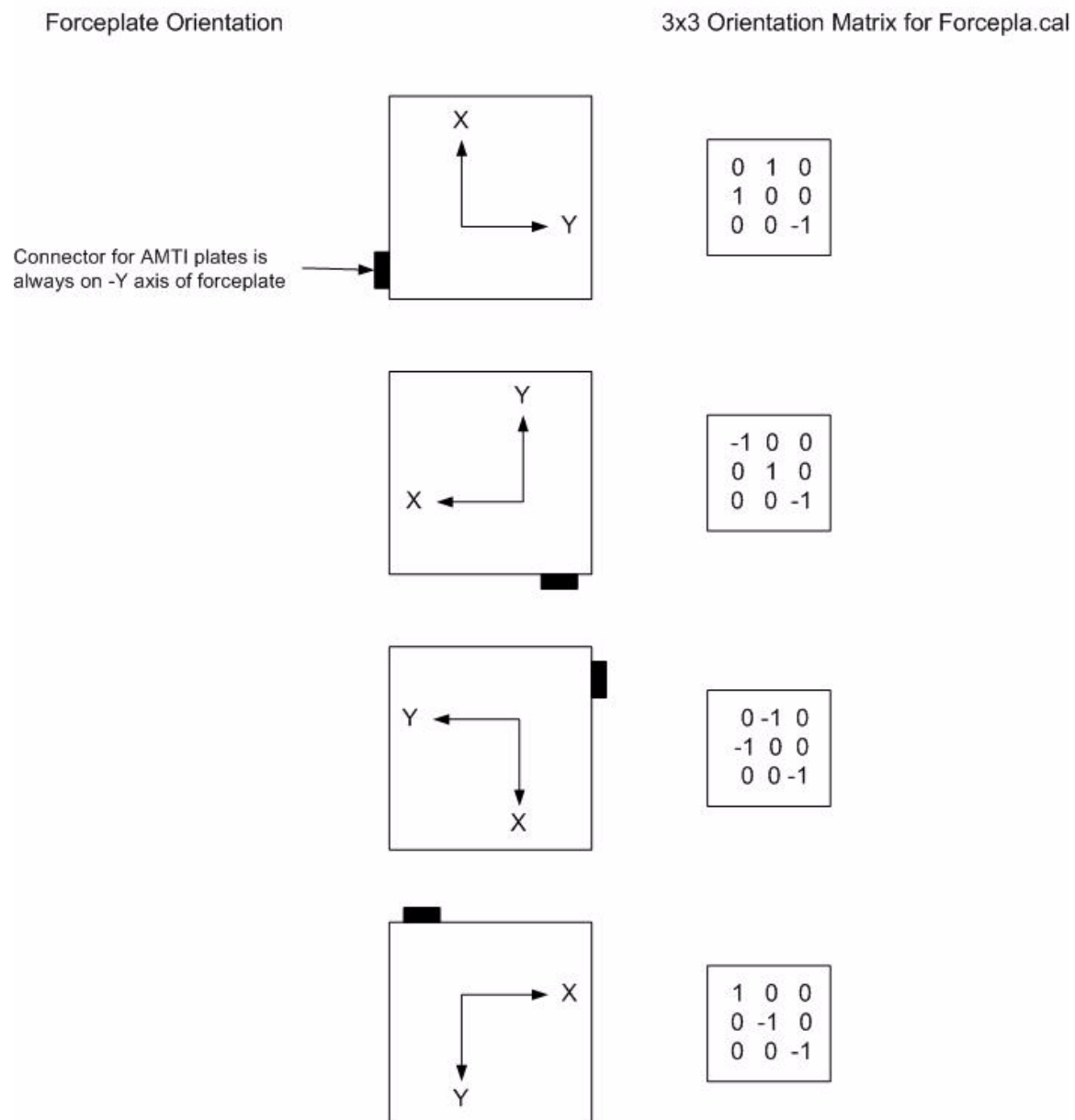
Forceplate Number	A unique number is assigned to each forceplate in the system.
Forceplate Scaling Factor and Optional Length & Width	The scaling factor depends on the forceplate manufacturer and the gain setting. Length and width are optional and are measured in cm. Length and width orientation is also dependent on the forceplate manufacturer. Refer to later sections specific to the manufacturer of your forceplate.
Forceplate Calibration Matrix	The calibration matrix transforms the output of the forceplate into forces and moments. Refer to the section later in this appendix specific to the manufacturer of your forceplate.
True XYZ Origin	This is the offset of the origin of the forceplate XYZ coordinate system relative to the center of the top surface of the forceplate. Each manufacturer provides this offset value.
XYZ Location in Video Coordinate System	This tells the Cortex system where the center of the top surface of the forceplate is located relative to the Cortex video coordinate system. Once this is established, the video calibration frame must be placed in the same location each time you calibrate. The center of the top surface can be found by measurement or drawing diagonal lines from opposite corners. The units of measurement are centimeters (cm).
3x3 Orientation Matrix	This matrix describes the orientation of the forceplate relative to the laboratory or room coordinate system. It is a matrix of direction cosines of the angles between the forceplate coordinate system and the laboratory coordinate system. Using the terminology $\cos(X_{\text{plate}}, X_{\text{lab}})$ to indicate the angle between the forceplate X axis and the laboratory X axis, the matrix takes the following form:

	X_{plate}	Y_{plate}	Z_{plate}
X_{lab}	$\cos(X_{\text{plate}}, X_{\text{lab}})$	$\cos(Y_{\text{plate}}, X_{\text{lab}})$	$\cos(Z_{\text{plate}}, X_{\text{lab}})$
Y_{lab}	$\cos(X_{\text{plate}}, Y_{\text{lab}})$	$\cos(Y_{\text{plate}}, Y_{\text{lab}})$	$\cos(Z_{\text{plate}}, Y_{\text{lab}})$
Z_{lab}	$\cos(X_{\text{plate}}, Z_{\text{lab}})$	$\cos(Y_{\text{plate}}, Z_{\text{lab}})$	$\cos(Z_{\text{plate}}, Z_{\text{lab}})$

Since, in real situations, the forceplate should be aligned with the room coordinate system, the numbers in this matrix will always have one of three values:

Angle = 0	cos = 1
Angle = 90	cos = 0
Angle = 180	cos = -1

Example matrices are shown in the following figure:

Figure E-3. Forceplate Coordinates System

Note: The video (lab) coordinate system is determined from the L-Frame orientation



Forceplate Scaling Factor, X-Width and Y-Length

The scaling factor depends on the forceplate manufacturer, and the forceplate amplifier gain setting and the voltage range.

Table E-1. Sample Forceplate Scaling Factors

Forceplate Manufacturer	Scaling Factor
AMTI	For Gain 4000—Use 25.0
Bertec	Use 0.5 if using AM6501 Analog Unit (output $\pm 5V$)
Kistler	± 10 Volt Amplifier—Use 1.0 ± 5 Volt Amplifier—Use 0.5

The x-width and y-length are the forceplate measurements in centimeters as measured in the forceplate coordinate system. Check the manufacturer's specifications. If no x-width and y-length values are used, AMTI and Bertec forceplates default to 18-inches by 20-inches, and Kistler forceplates default to 50-centimeters by 50-centimeters.

Using AMTI and Bertec Forceplates

AMTI Gain Setting

For the AMTI forceplates, a gain of 4000 mV and a cutoff frequency of 1050 kHz is recommended. Using the method outlined in the AMTI literature, this gain yields a scaling factor for the **forcepla.cal** file of 25. The example in [Figure E-4](#) uses an AMTI forceplate.

Bertec Gain Setting

A gain setting of 10 for Bertec forceplates is recommended. In the **forcepla.cal** file you should set *scaling factor* = $1 / \text{gain}$, yielding a **scaling factor** of 0.1. If using the AM6501 analog out device, the gain is preset and the analog voltage range should be $\pm 5\text{V}$.

A scaling factor of 0.5 is recommended in the forcepla.cal file.

The Calibration Matrix

The 6x6 calibration matrix (Inverted Sensitivity) is provided by the manufacturer (AMTI). It is used to transform the output of the force-plate into three force vectors and three moment vectors. The form of the matrix is shown in [Figure E-5](#). The main diagonal of the matrix (upper left to lower right) represents the basic channel sensitivities. The off diagonal terms represent the channel cross-talk.

If using Bertec plates with the AM6501, the amplifier is built into the plate and outputs data using a 15-pin D-Sub-connector cable to a small interface unit with signals in the range $\pm 5\text{V}$. In this case the amplifier is pre-calibrated, internally, with no off-axis elements in the 6x6 matrix. There are pre-defined and fixed values for the diagonal axis values. These can be exactly the same for each plate but check the Bertec documentation for the values appropriate to your plate. See the sample forcepla.cal file ([Figure E-4](#)). If using other Bertec hardware (e.g. AM6504 or AM6800), please check the output range and scale factors with the manufacturer.

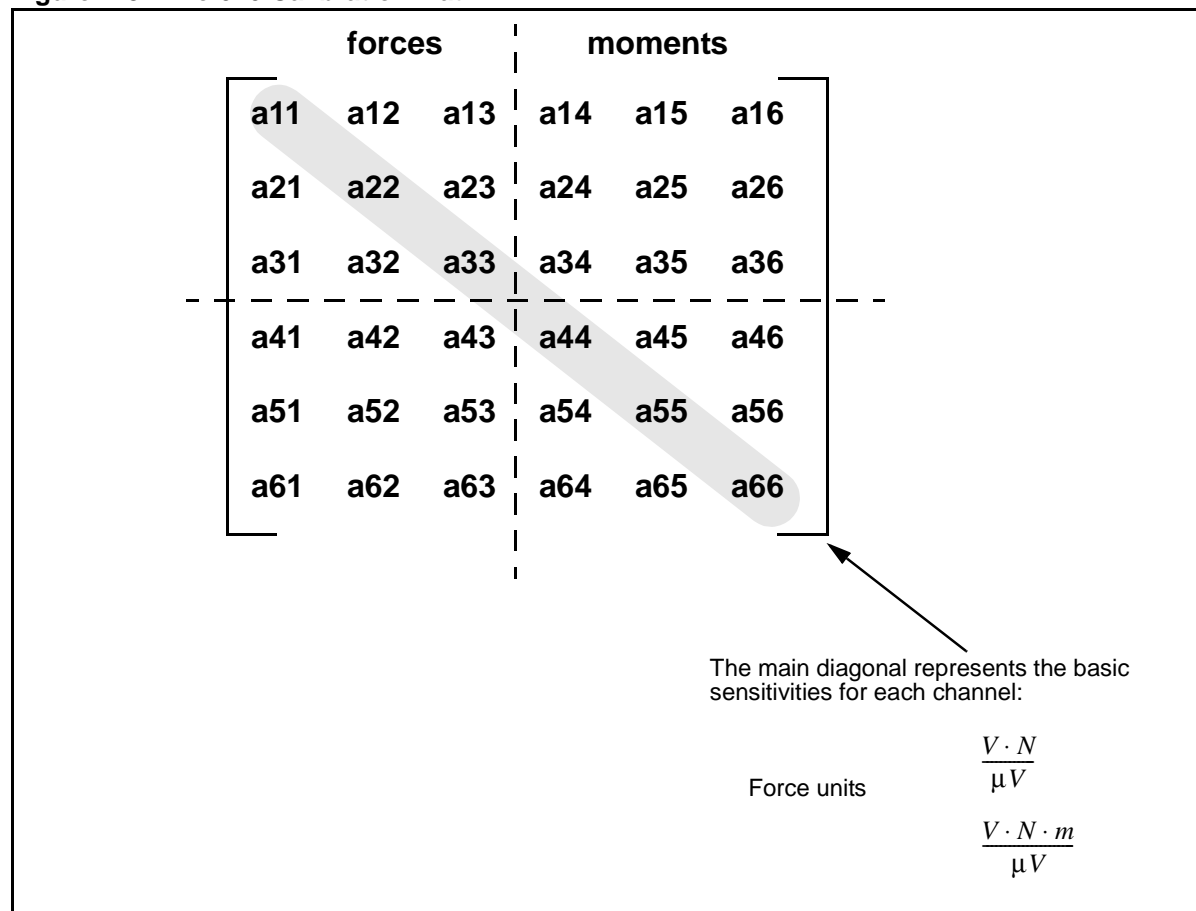
Figure E-4. Example Forcepla.cal File for 2 Forceplates

```

1.
25 51 46.5
2.9350 0.0040 0.0130 -0.0480 -0.0400 0.0050
-0.0020 2.9930 0.0470 0.0080 -0.0250 0.0410
-0.0270 0.0120 11.5420 -0.0240 0.0030 0.0160
0.0000 -0.0070 0.0000 1.5390 -0.0110 0.0010
-0.0070 -0.0020 0.0000 0.0000 1.5350 -0.0040
0.0020 0.0020 -0.0050 0.0020 -0.0040 0.7440
-0.1000 -0.0260 -3.8000
5.6000 -25.7000 -4.2000
0.0000 1.0000 0.0000
1.0000 0.0000 0.0000
0.0000 0.0000 -1.0000
2.
25 51 46.5
2.9340 0.0090 0.0020 -0.0130 0.0170 0.0110
0.0120 2.9750 0.0450 0.0310 -0.0200 0.0340
0.0050 0.0020 11.5480 -0.0210 -0.0090 0.0070
-0.0040 0.0010 0.0000 1.5470 -0.0020 -0.0030
0.0010 -0.0020 0.0000 0.0020 1.5450 -0.0080
0.0020 0.0060 0.0000 0.0010 -0.0060 0.7480
-0.1000 0.0300 -4.2000
56.5 -25.7000 -4.2000
0.0000 -1.0000 0.0000
-1.0000 0.0000 0.0000
0.0000 0.0000 -1.0000

```

Figure E-5. The 6x6 Calibration Matrix



The information provided by the manufacturer may include only the basic sensitivities for each channel with no values for cross-talk. In this case, the matrix should be filled with the basic sensitivities on the main diagonal and zeroes elsewhere.

Also notice that the upper right quadrant of the matrix contains the force sensitivities and the lower right contains the moment sensitivities. In every case, the force sensitivities are greater than the moment sensitivities. **Cortex** uses this information to switch matrix quadrants (permute the matrix) if the manufacturer should supply the matrix with the moments on the left and force on the right.

Note: The calibration matrix is intended to be used with your plate's coordinate system, not the room's. For this reason, if your plate is not aligned with the room, correct it with the 3x3 orientation matrix, not by switching wires or A/D signal names.

Using Kistler Forceplates

Signal Names

The Kistler forceplate has 8 outputs. Therefore, two forceplates will use 16 channels on the A/D card. The signal naming conventions are shown in [Figure E-6](#). The names in the analog ANB (or ANA) file must appear exactly as shown in the **Cortex** ANB (or ANA) column.

Gain Setting

With the Kistler forceplate, the Charge Amplifier (model 9865) should be set on **range #3** for the X/Y and Z range settings (X and Y are set together). This is the 10,000 pC setting.

This setting can be changed if desired, but the **forcepla.cal** file will have to reflect the change. A **gain = 1** on the A/D board should be used since the Kistler outputs 10 V full scale.

Calibration Matrix

The Kistler forceplate requires an 8x8 calibration matrix. The matrix only contains non-zero data on the main diagonal (upper left to bottom right). All non-diagonal cross-talk elements are zero.

To calculate the values to use on the main diagonal of the matrix (assuming nominal sensitivity values of 7.8 and 3.8 pC/N):

$$\text{X and Y Scaling} \quad (10000 \text{ pC} / 7.8 \text{ pC/N}) / 10 \text{ V} = 128.2 \text{ N/V}$$

$$\text{Z Scaling} \quad (10000 \text{ pC} / 3.8 \text{ pC/N}) / 10 \text{ V} = 263.4 \text{ N/V}$$

[Figure E-6](#) shows an example 8x8 matrix in a **forcepla.cal** file.

True XYZ Origin

This is a measure of the X, Y, and Z distances to the piezoelectric transducers used to generate the signals in the Kistler forceplates. These numbers are supplied by the manufacturer.

Figure E-6. Example Forcepla.cal File For a Kistler Forceplate

1K							
1.0							
128.2	0	0	0	0	0	0	0
0	128.2	0	0	0	0	0	0
0	0	128.2	0	0	0	0	0
0	0	0	128.2	0	0	0	0
0	0	0	0	263.4	0	0	0
0	0	0	0	0	263.4	0	0
0	0	0	0	0	0	263.4	0
0	0	0	0	0	0	0	263.4
12.000	20.000	-5.4000					
0.0	0.0	0.0					
1	0	0					
0	-1	0					
0	0	-1					

General Notes On Kistler Forceplates

1. Since the Kistler forceplate format is flagged with a “K” after the forceplate number, Kistler and other forceplates may be included in a single system.
2. The proper way to orient the forceplate is the 3x3 orientation matrix, not the calibration matrix.

Note: Do not switch the cables to the A/D board.

3. Keep the Long Term Constant turned off on the charge amplifier.
4. Reset the charge amplifier before each test, or at least every few tests. This re-establishes the zero for the charge amplifier.

Using Kyowa Dengyo Forceplates

The Kyowa Dengyo force plate has now been incorporated into **Cortex**.

The following is a description of the procedure for calibration using the RealTime interface as well as a description of the **forcepla.cal** for Kyowa Dengyo force plates.

The automatic calibration of the Kyowa Dengyo force plates is now implemented in **Cortex**. At the end of the calibration procedure the Real Time system creates a new **forcepla.cal** file containing the latest calibration values (zero, +cal and –cal) for computing the distortion conversion coefficient. The calibration procedure is as follows:

1. Make sure that the Kyowa Dengyo force plates are connected to the National Instruments A/D data acquisition hardware with the following channel assignments:

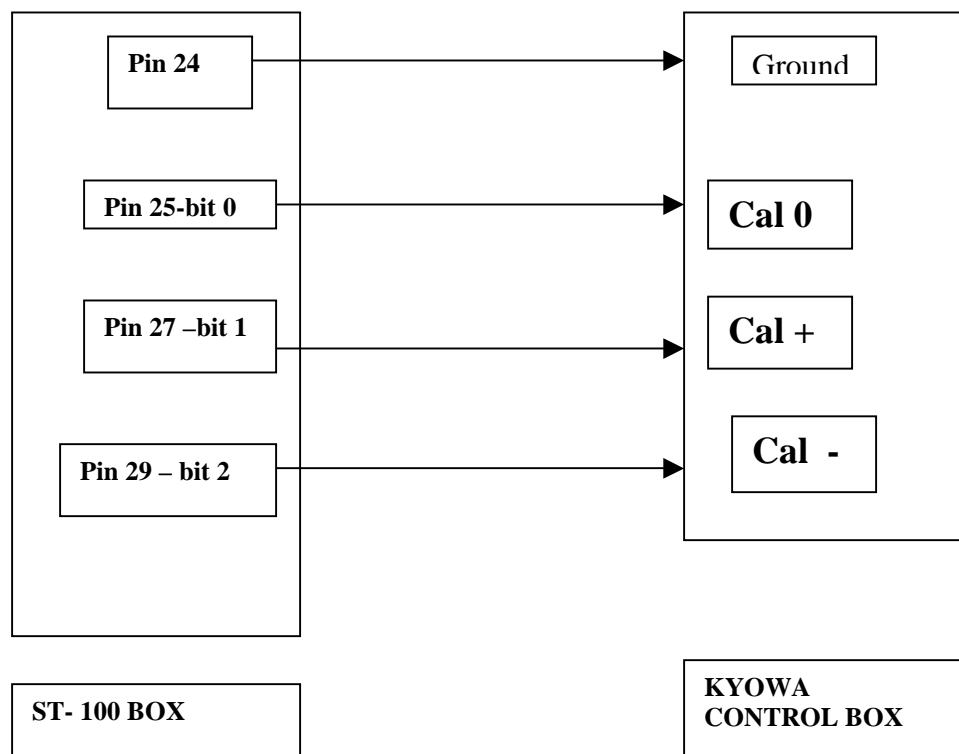
Table E-2. Force plate channels for Kyowa Force Plate number 1

Analog Channels	Kyowa Dengyo Force Plate Channels
Channel 1	FZ11
Channel 2	FZ12
Channel 3	FZ13
Channel 4	FZ14
Channel 5	FX114
Channel 6	FX123
Channel 7	FY112
Channel 8	FY134

Follow a similar connection and naming (FZ21, FZ22 etc.) sequence for additional plates.

2. Connect the ST-100 box terminal pins 25, 27, and 29 (pin 24 is ground) which correspond to bits 0, 1 and 2 of the digital I/O of the National Instrument A/D board to Cal (o), Cal (+) and Cal (-) of the Kyowa Dengyo interface box.

Figure E-7. Kyowa Connection Block Diagram



3. The Folder containing the **Cortex** executables must also have a **force-pla.cal** file (previously created) according to the format for **force-pla.cal** file for Kyowa Force Plates.

Load a project file that has the Kyowa forceplates enabled in the Analog set up panel

After making sure all the connections are properly made click on the **Calibrate Kyowa Force Plates** button.

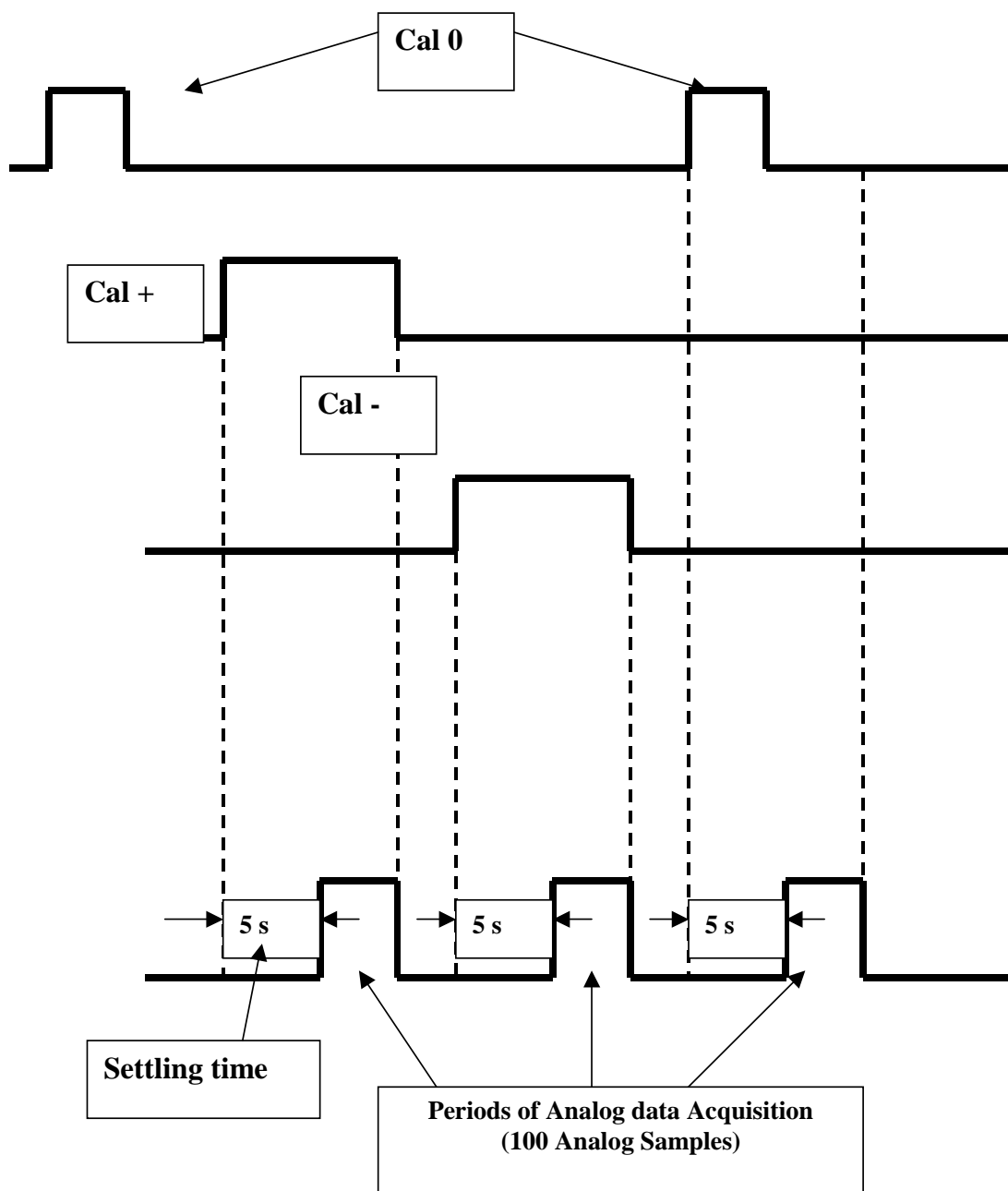
The following sequence of events are initiated.

- TTL pulses are sent, in sequence to CAL 0, CAL+, CAL- and CAL ZERO terminals of the Kyowa control box as shown on the timing diagram.
- 100 samples of analog data are collected across all the Kyowa force plate channels, 5 seconds after the initiation of the TTL pulses.
- Approximately 1 minute is required for the calibration.

- The average of the 100 samples (in A/D units) are computed and data are written into a **forcepla.cal file** in the same folder as the loaded project file.

The timing sequence of the TTL pulses and the analog data acquisition is shown on the next page.

Figure E-8. Timing Diagram for Kyowa Force Plate Calibration



**Description of
Forcepla.cal File
for Kyowa
Dengyo
Forceplates**

[Line 1] [Force plate number Kyowa] Example:1Kyowa, 2Kyowa etc.,.

[Line 2] [Scale factor Width Length]

[Line 3] [Calibration Range Settings in the order FZ1 FZ2 FZ3 FZ4
FX14 FX23 FY12 FY34]

[Line 4] [Zero values in A/D units written by Cortex Calibration step]

[Line 5] [Cal + values in A/D units written by Cortex Calibration step]

[Line 6] [Cal - values in A/D units written by Cortex Calibration step]

[Line 7] [Load Conv.Coeff.*Voltage Conv. Coeff.*9.801] same order as
above

[Line 8] [XY axis Conversion Coefficients(XY locations of Z-axis force
Transducers) in centimeters]

[Line 9] [X location of Fy and Y location of Fx transducers in centime-
ters]

[Line 10] [Location of Geometric Center of Force Plate with respect to
video coordinate origin in video coordinates]X Y Z in Centimeters

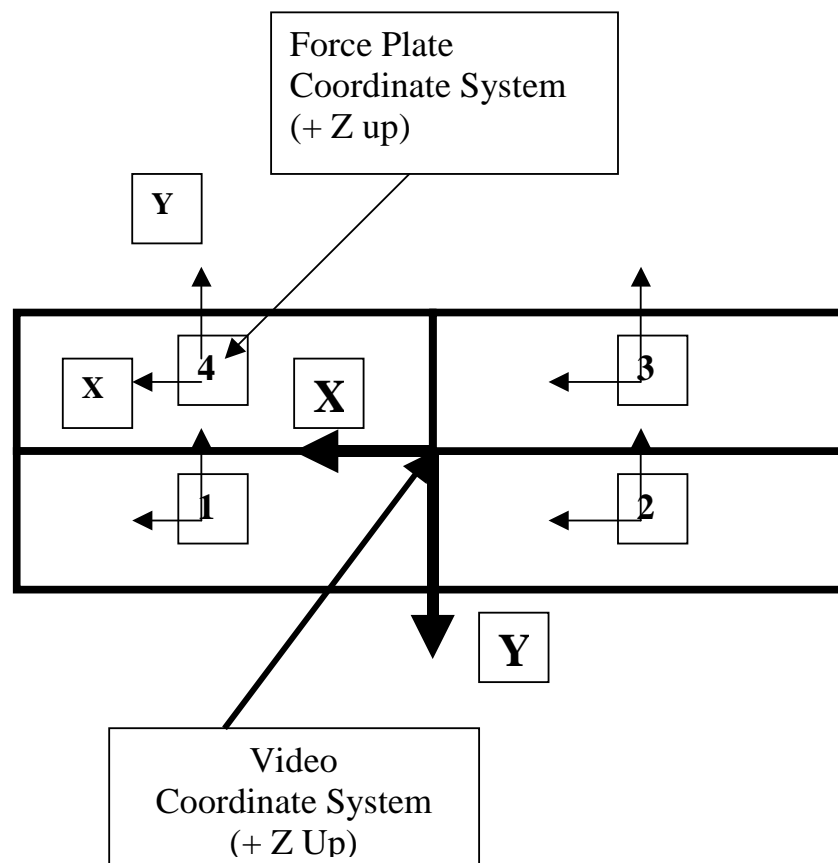
[Line 11]

[Line 12] [Force Plate Orientation Matrix]

[Line 13]

Example Kyowa Dengyo Forcepla.cal File

Figure E-9. Example Forcepla.cal file for 4 Kyowa Dengyo Force Plates



```

1Kyowa
1.0  60  180
2000      2000      2000      2000      500      500
500      500
-3    2    -4    3    6    -6    5    -6
1600  1601  1602  1599  803  802  800  802
-1608 -1609 -1608 -1608 -805 -804 -804 -805
0.00103 0.00104 0.00103 0.00103 0.00073 0.00073 -
0.00071 -0.00071
-22.5017 -22.3307 22.8603 22.1836 55.7321 -55.2022
-54.9477 54.5368
23.35 51.00
90.0 30.0 0.00
0.0000 1.0000 0.0000
-1.0000 0.0000 0.0000
0.0000 0.0000 1.0000
2Kyowa
1.0  60  180
2000      2000      2000      2000      500      500
500      500
-8    5    -6    3    -22  26    -8    -11
1609  1606  1606  1607  803  803  805  803
-1608 -1607 -1605 -1607 -802 -804 -803 -802
0.00101 0.00102 0.00101 0.00101
0.00079 0.00072 -0.00074 -0.00075
-22.5648 -22.3288 22.3755 22.7563
54.8645 -55.3299 -55.3140 55.7631
23.35 51.00
-90.0 30.0 0.0
0.0000 1.0000 0.0000
-1.0000 0.0000 0.0000
0.0000 0.0000 1.0000
3Kyowa
1.0  60  180
2000      2000      2000      2000      500      500
500      500
-21  25  24  22  -18  21  -10  15
1613  1606  1605  1607  805  806  800  804
-1610 -1605 -1601 -1606 -803 -806 -808 -804
0.00102 0.00103 0.00101 0.00104
0.00072 0.00072 -0.00075 -0.00075
-22.5481 -22.3083 22.3729 22.7725
55.2674 -54.6867
-55.6732 55.1962
23.35 51.00

```

```

-90.0      -30.0      0.00
0.0000     1.0000     0.0000
-1.0000     0.0000     0.0000
0.0000     0.0000     1.0000
4Kyowa
1.0   60   180
2000   2000   2000   2000   500   500
500   500
-0    2    2    4    10   -3    6    -5
1602 1607 1604 1603 801   805   803   803
-1608 -1614 -1612 -1610 -805 -808 -807 -806
0.00104 0.00104 0.00102 0.00103
0.00072 0.00073 -0.00071 -0.00072
-22.3702 -22.5673 22.6288 22.4559
55.3667 -55.2103 -54.9972 55.2328
23.35 51.00
90.0      -30.0      0.00
0.0000     1.0000     0.0000
-1.0000     0.0000     0.0000
0.0000     0.0000     1.0000

```

Appendix F ***SDK—Software Developers Kit***

Topic	Page
SDK Overview	F-1
SDK Programming Example: Write your own Streaming Plugin	F-1

SDK Overview

The SDK is available for the advanced user who wishes to incorporate the output data stream from **Cortex** into a software application.

The Software Developers Kit, which provides the tools for interfacing your program with **Cortex** is available by special request from Motion Analysis Corp. at Support@MotionAnalysis.com.

SDK Programming Example: Write your own Streaming Plugin

There is a Software Development Kit (SDK) which is written in the Visual C/Visual C++ language and an example C program that is available at no charge that demonstrates how to use the SDK. This allows our customers to use this as a starting point and creating their own program.

The sample program shows you how to connect to the **Cortex** software and request that the kind of data that you want be transferred (Marker XYZ data and/or Skeleton HTR data). The sample C program then writes out the data to a disk file. Please also note that the data can be streamed either from the live camera data (that is happening in real time), or from the Post Processing software when you press the Play button. The data is the same either way and the SDK program does not even need to know since it comes across the same way. So, the customer can write the program from previously edited XYZ data in the Post Processing part of **Cortex** and get that to working. Press the **Play** button, you see the edited tracks and the data is streamed to the SDK. Then they can connect to the cameras and get the same XYZ or HTR type data from the live cameras.

Appendix G *Import and Export File Formats*

Topic	Page
Overview	G-1
mac_lic.dat	G-2
PRJ—Cortex Project File	G-3
TRC—Track Row Column	G-3
HTR2—Hierarchical Translations and Rotations	G-5
HTR	G-9
ANC—Analog ASCII Row Column	G-15
TS—Time Series Files from the Cortex Analysis Functions: Velocity and Acceleration Calculations	G-16
Binary Files—ANB, TRB, and C3D	G-18

Overview

The files generated by **Cortex** fall into two main categories: ASCII and binary. ASCII files contain data in a text form that can be read by any text editor. They usually have descriptive headers that indicate the nature of the data that follows. Often these files are in a row and column format that allows data to be read and manipulated by spreadsheet programs such as **Excel**[™]. ASCII files are not compact and can be quite large.

Binary files contain raw binary data and are more compact than ASCII files. They cannot be read by a text editor. In general, binary files are not meant to be read by the end user.

mac_lic.dat

All Motion Analysis software requires a valid license to run. The license is keyed to a particular computer.

- For a **Windows NT** computer, it is keyed to the number of the dongle supplied by Motion Analysis.
- For **SGI** computers, it is keyed to the sysinfo-s number.
- For **Sun/Sparc** computers, it is keyed to the host id number.

The license file is ASCII and has the unique name **mac_lic.dat**. It is located in the **Motion Analysis** directory. Only one license file is used for all Motion Analysis software. Each additional software application beyond **Cortex** is given a separate line in the license file, with the license type enclosed in square brackets [] followed by two license codes.

If you acquire new Motion Analysis software, you must use a text editor to enter the new line in your license file to enable your new software. This line can be typed in or entered by cutting it from the file you receive and pasting it into the existing license file. The order of the licenses in the file is not important and only those lines that start with a [(left bracket) are read by the software.

Note: If you are operating in Windows XP or Vista, make sure that file extensions are not hidden, which is the default. This makes the **mac_lic.dat** look like **mac_lic**, which might be renamed to **mac_lic.dat.dat**. If this happens, the system will not recognize the license file.

Figure G-1. An Example of a Motion Analysis License File

```
Motion Analysis License File
Customer: MAC Customer
Platform: NT
SystemID: 19c
Created: 9/15/20xx 1:42:26 PM
Sales Order#: 05-xxx
Entered By: Support
[Cortex 1.0] aed50167      873b2d56
[Analog Input]      b9806c31      d1567841
[OrthoTrak]         b2df5e69      8964274a
[Animation Plugins] b1a46160      805b5c49
[Director/Sequencer] e1a04e65      85745819
[RT2 Animation Plugins] e3f05340      a069081b
[Analog Input]      b9806c31      d1567841
[Calcium 4]         e7ed5923      c363151f
[Skeleton Builder 4] a3f44279      99780c5b
[Reference Video 3.0] eb92592f      cf636a13
[Talon Streaming 4]  ecb36136      d65b4b14
[Talon Viewer 4]     86fb0714      f43d037e
[Motion Composer]    c7f00e25      c534083f
This license has no expiration.
```

PRJ—Cortex Project File

Every motion capture session must have a project file containing all system settings, equipment parameters, and other information related to the project. This file contains both equipment parameters common to many different setups and calibration values unique to one particular session. Among the data found in a project file are:

- the system setup
- the marker set
- calibration setup and results
- linkages between markers
- SkB segment definitions, coordinate systems, and hierarchies
- (optional) MoCap Solver segment definitions, joint types, and hierarchies
- camera type and parameters

In most cases, you will begin a session by loading an existing project file, editing it as necessary, and saving it in the directory where the motion data is to be saved. Any time you calibrate the system or edit project parameters, you should save the project file to disk to retain the new information.

Important

Project files contain ASCII data and it may be useful to view them using any text editor, however, you should never edit them in a text editor.

TRC—Track Row Column

The **.trc** file contains X-Y-Z position data for the reflective markers. This is an ASCII file in a Row/Column, horizontal tab delimited format that can be easily read into a spreadsheet program such as **Excel™** and **Lotus™**. The position data for each marker is organized into 3 columns per marker (X, Y and Z position) with each row being a new frame. The position data is relative to the global coordinate system of the capture volume and the position values are in the units used for calibration.

The file is made up of three parts:

- the file header,
- the position data header, and
- the position data.

All fields in this file type are separated by horizontal tabs.

File Header

The **.trc** file header occurs on the first three rows.

- Row one contains the path file type label (string), path file type number (int), path file type descriptor (string) and original directory path and file name (string).

- Row two contains the data rate label (string), the camera rate label (string), the number of frames label (string) and the units label (string).
- Row three contains the data rate value (real), the camera rate value (real), the number of frames (int) and data units (string).

Data Header

The data header occupies rows four and five.

- Row four contains the frame number label (string), the time label (string) and followed by the marker name labels (string). There are three horizontal tab characters between each marker name label. These names usually correspond to the location where a reflective marker was placed on the subject.
- Row five contains the column labels (string) for the position data starting on row six. For each marker name there is an X, Y and Z column. These axes labels have the trajectory numbers appended to them.

Position Data

Position data begins at row six. Column one of the data fields contain the frame number (int). Column two contains the time (real) and columns from three on contain the X, Y and Z position data (real) for each trajectory. There are three columns for every trajectory.

Empty Fields

An empty frame of position data (missing data) is represented as three consecutive horizontal tab characters.

Example

Shown below is a portion of a file with the following attributes:

- captured rate = 60 frames per second
- total frames = 90
- total reflective markers = 33
- units of measure = mm

Data for only the first 3 markers is shown, the remaining markers would appear in columns to the right. Also, data for only the first 12 frames is shown, the remainder would appear in rows below frame 12.

Figure G-2. An Example of a TRC File

PathFileType4 (X/Y/Z) /usr/people/evademo/Oct14/MichelleInit1.trc											
DataRate	CameraRate	NumFrames			NumMarkers			UnitsOrigDataRate			
.60.0	60.0	90			33			mm	60.0		
Frame#Time	Head_Top	LHead						RHead			.
		X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3	.
1	0.000	234.5437	1673.7619	232.2308	316.7533	1608.3785	218.5500	144.7963	1597.3691	274.4994	.
2	0.017	235.2399	1673.4542	232.1284	316.6074	1608.2884	219.0597	144.7684	1597.8106	274.2137	.
3	0.033	235.2361	1673.4926	232.1852	316.1265	1608.1984	219.7480	144.7296	1597.8043	274.3916	.
4	0.050	235.0781	1673.4376	232.2152	316.5659	1607.9659	219.6344	144.7652	1597.5711	274.4420	.
5	0.067	235.0781	1673.4376	232.2152	316.7533	1608.3785	218.5500	144.7296	1597.8043	274.3916	.
6	0.083	235.0781	1673.4376	232.2152	316.2800	1608.3460	219.1676	144.7963	1597.3691	274.4994	.
7	0.100	235.3844	1673.4179	232.3085	316.4539	1608.1409	219.6402	144.7963	1597.3691	274.4994	.
8	0.117	235.1416	1673.4447	232.4143	316.4539	1608.1409	219.6402	145.1222	1597.3621	274.3569	.
9	0.133	235.4312	1673.4882	232.7111	316.1265	1608.1984	219.7480	144.7963	1597.3691	274.4994	.
10	0.150	236.0334	1673.4269	233.0335	315.9777	1608.3370	220.0773	144.6867	1597.5786	274.4743	.
11	0.167	235.7562	1673.7265	233.4352	316.4172	1608.1043	219.9638	144.7963	1597.3691	274.4994	.
12	0.183	235.5336	1673.4414	233.5973	315.9770	1608.0798	220.2932	144.7963	1597.3691	274.4994	.
.

HTR2—Hierarchical Translations and Rotations

The HTR2 (*.htr2) file contains rotations (about X, Y and Z axis) for the body segments defined in the **Cortex** Project and translations and rotations for the root segment. Rotations are calculated relative to a local coordinate system of each segment's designated parent.

The HTR files translation are expressed in the units used for **Cortex** system calibration and the rotations are calculated as Euler angles expressed in degrees. These Euler angles are either bounded or continuous.

- **Bounded**—indicates that when the angles are extracted they are bounded or constrained between ± 180 degrees for the X and Z directions, and ± 90 degrees for the Y direction.
- **Continuous**—means that the angles will be continuous, i.e. the angles are not bounded. With unbounded angles you can conceivably have an angle that goes from 0 to 1,000 degrees for each one of the X, Y and Z angles.

This HTR file has four main parts: the Header; the Segment Names and Hierarchies, the Base Position, and the Data.

Typically, the Base Position frame is selected when the subject's body is in a symmetrically oriented, neutral stance position. This Base Position frame is very important because this will be the position and orientation

where each segment's translation and rotation is set to zero and the bone length scale factor is set to 1.0.

Since all segments are hierarchical, child segments have their translations and rotations relative to their parent. The origin of a child segment is found by applying the translations relative to the parent's coordinate system. The orientation of the child segment can be established by rotating the child's coordinate system relative to the parents coordinate system about each of the axes.

In the data section, the order of transformation is: translation followed by rotation. The segment names are keywords, for example **head**. Each segment's data is contained in seven columns, translations in X, Y and Z, rotations about X, Y and Z and a scale factor for the Y axis or bone length. Each frame of data is represented on one row.

Example

The complete file contains data for each of the 20 segments. The position of each segment is recorded for 387 video frames. Such a file is quite large, so we have included an abbreviated version here.

The file begins with a **[Header]** section containing general information, such as the number of segments, the number of frames, the frame rate, and other parameters which apply to all data in the file.

This is followed by the **[SegmentNames&Hierarchy]** section which describes the child-parent relationships of the skeleton. Notice that only the **LowerTorso** segment relates to the **GLOBAL** coordinate system. All other segments motions are described in relation to a parent segment.

In the **[BasePosition]** section, the location of each segment's origin and rotation are described in the skeleton's base position, using the six available degrees of freedom:

Translation in X	Rotation about X axis
Translation in Y	Rotation about Y axis
Translation in Z	Rotation about Z axis

For the child segments, location and rotation are given in terms of the parent. In this skeleton, the origin of all the children lie near the bone (Y axis) of the parent and, therefore, have only Y values. The seventh column gives the length of the bone segment.

The remaining sections contain motion data for each segment, in each frame, relative to the base position. This is frame oriented, meaning each section holds all segments for that particular frame.

In this abbreviated example, only the first and last four frames are shown for the first three segments. In the actual file, all 387 frames for each of the 20 segments would appear. After all the segments, an **[EndOfFile]** section terminates the file.

Figure G-3. An Example of an HTR2 File

```

# Hierarchical Translation and Rotation (.htr) file
# Generated by Cortex
[Header]
FileType htr
DataType HTRS
FileVersion 2
NumSegments 20
NumFrames 511
DataFrameRate 60
EulerRotationOrder ZYX
CalibrationUnits mm
RotationUnits Degrees
GlobalAxisofGravity Y
BoneLengthAxis Y
ScaleFactor 1
[SegmentNames&Hierarchy]
#CHILD          PARENT
Head            Neck
Neck            UpperTorso
UpperTorso      LowerTorso
LCollarBone     UpperTorso
RCollarBone     UpperTorso
LUpArm          LCollarBone
RUpArm          RCollarBone
LLowArm         LUpArm
RLowArm         RUpArm
LHand           LLowArm
RHand           RLowArm
LowerTorso      GLOBAL
LPelvis         LowerTorso
RPelvis         LowerTorso
LThigh          LPelvis
RThigh          RPelvis
LLowLeg         LThigh
RLowLeg         RThigh
LFoot           LLowLeg
RFoot           RLowLeg
[BasePosition]
#SegmentName    Tx   Ty   Tz   Rx   Ry   Rz   BoneLength
Head            0.0  0.0  0.0  0.0  0.0  0.0  1.0
Neck            0.0  0.0  0.0  0.0  0.0  0.0  1.0
UpperTorso      0.0  0.0  0.0  0.0  0.0  0.0  1.0
LCollarBone     0.0  0.0  0.0  0.0  0.0  0.0  1.0
RCollarBone     0.0  0.0  0.0  0.0  0.0  0.0  1.0
LUpArm          0.0  0.0  0.0  0.0  0.0  0.0  1.0
RUpArm          0.0  0.0  0.0  0.0  0.0  0.0  1.0
LLowArm         0.0  0.0  0.0  0.0  0.0  0.0  1.0
RLowArm         0.0  0.0  0.0  0.0  0.0  0.0  1.0
LHand           0.0  0.0  0.0  0.0  0.0  0.0  1.0
RHand           0.0  0.0  0.0  0.0  0.0  0.0  1.0
LowerTorso      0.0  0.0  0.0  0.0  0.0  0.0  1.0
LPelvis         0.0  0.0  0.0  0.0  0.0  0.0  1.0
RPelvis         0.0  0.0  0.0  0.0  0.0  0.0  1.0
LThigh          0.0  0.0  0.0  0.0  0.0  0.0  1.0
RThigh          0.0  0.0  0.0  0.0  0.0  0.0  1.0
LLowLeg         0.0  0.0  0.0  0.0  0.0  0.0  1.0
RLowLeg         0.0  0.0  0.0  0.0  0.0  0.0  1.0
LFoot           0.0  0.0  0.0  0.0  0.0  0.0  1.0
RFoot           0.0  0.0  0.0  0.0  0.0  0.0  1.0
#Beginning of Data.

```

Frame 1:

0:	265.65848	958.52289	171.77657	
1:	-5.56661	-1.03833	-3.08909	222.84282
2:	6.64226	-10.44067	-1.80951	81.65315
3:	-5.68993	-4.37941	2.25734	279.37607
4:	0.21077	8.22242	-101.23232	148.15902
5:	-2.91121	-14.64063	98.93079	122.55443
6:	7.62039	4.41526	6.57253	286.27760
7:	4.45230	6.92158	-1.13849	273.89803
8:	0.27224	1.42203	2.06549	197.27960
9:	10.70936	0.37209	-6.77907	227.77268
10:	5.60677	0.99909	-5.37607	160.71960
11:	-7.93797	-1.63432	4.66433	138.17473
12:	7.92266	6.08590	0.69348	116.27502
13:	17.83143	2.88770	-106.49151	122.39289
14:	5.29669	4.26448	108.76140	124.66989
15:	3.41580	-26.21330	-60.12212	448.80791
16:	0.59471	18.20674	59.84934	469.08647
17:	-6.65161	-0.51561	-9.07148	349.00357
18:	-8.69699	2.51242	6.54966	359.40872
19:	56.16376	-3.45440	1.11591	184.70566
20:	69.34235	4.62345	-3.69895	181.04145

Frame 2:

0:	265.98917	958.55890	171.80534	
1:	-4.76026	-0.93315	-1.37413	222.40665
2:	6.24591	-10.44039	-3.43725	81.65838
3:	-5.45918	-4.48315	2.16439	279.15316
4:	0.25053	8.10135	-101.08283	148.41689
5:	-2.74117	-14.49931	99.00355	122.50407
6:	7.56453	4.56852	6.41734	286.17036
7:	4.44328	6.91024	-1.35979	273.86227
8:	0.13121	1.30640	2.05696	196.92114
9:	10.54723	1.02512	-6.42520	227.48704
10:	6.09020	1.13973	-5.56063	160.93207
11:	-7.83955	-2.10847	4.52455	138.30477
12:	7.76307	6.08555	0.77588	116.44541
13:	17.85974	3.07533	-106.59285	122.55485
14:	5.44536	3.99245	108.62376	124.97174
15:	3.05847	-26.22045	-60.08476	448.35952
16:	0.34028	16.47090	59.49600	468.76412
17:	-6.65867	-0.44105	-9.10057	349.54976
18:	-8.45317	4.60744	7.28953	359.82323
19:	56.25188	-3.41367	1.03058	184.85654
20:	69.34779	4.49733	-4.24123	181.27314

.....

.....

[EndOfFile]

HTR

HTR files provide the same information as HTR2 files, but in a segment oriented method which is less suitable for information streaming. Refer to [“HTR2—Hierarchical Translations and Rotations” on page G-5](#). The HTR format cannot be created in **Cortex**. This feature will be added to future revisions.

Examples of HTR Files

Information about the structure and motion of hierarchical skeletons is stored in **.htr** files. There are two variations of **.htr** files: version 1 (HTR) and version 2 (HTR2). The skeleton data is identical in both file versions, however, the motion data is presented on a segment basis in version 1 files, while it is on a frame basis in version 2 files. Thus, version 1 files give the position data for all frames for the first segment followed by the position data for all frames for the second segment, etc. HTR Version 1 files are used:

1. to save as HTR in **Cortex** under the Post Skeleton function in the Tools menu
2. for input and output from **Si 2.0**
3. for input and output for many of the animation package software

HTR2 files are output from streaming mode from **Cortex 3.0** and **Cortex**, but there is no software to import them. Version 2 file gives the position data for all segments for the first video frame followed by the position data for all segments for the second frame, etc.

Example of an HTR Version 1 File

The example of a version 1 file shown in [Figure G-4 on page G-11](#) was generated by Motion Analysis **MoCap Solver** and contains data for the movement of a hierarchical skeleton with one root and 19 child segments. Since the file was generated by **MoCap Solver**, the skeleton has fixed length bones.

The complete file contains data for each of the 20 segments. The position of each segment is recorded for 196 video frames. Such a file is quite large, so we have included an abbreviated version here.

The file begins with a **[Header]** section containing general information, such as the number of segments, the number of frames, the frame rate, and other parameters which apply to all data in the file.

This is followed by the **[SegmentNames&Hierarchy]** section which describes the child-parent relationships of the skeleton. Notice that only the **LowerTorso** segment relates to the GLOBAL coordinate system. All other segments motions are described in relation to a parent segment.

In the **[BasePosition]** section, the location of each segment's origin and rotation are described in the skeleton's base position, using the six available degrees of freedom:

Translation in X	Rotation about X axis
Translation in Y	Rotation about Y axis
Translation in Z	Rotation about Z axis

For the child segments, location and rotation are given in terms of the parent. In this skeleton, the origin of all the children lie on the bone (Y axis) of the parent and, therefore, have only Y values. The seventh column gives the length of the bone segment.

The remaining sections contain motion data for each segment relative to the base position. Each section starts with the segment name followed by position data and the scale factor (SF) for each frame. The first segment, **[LowerTorso]**, is unique because it has six values describing its relation to the global coordinate system. All subsequent segments have only three rotational degrees of freedom. Therefore, if you wished to find the global coordinates of any segment in any frame you would follow these steps:

- Calculate the three rotation values for each segment using the values from the desired frame and the base position for that segment. For the root, also calculate each of the three translation values. Be careful to use the correct rotation order as indicated in the header of the **.htr** file.
- Using the positions and rotation of the root segment as a starting point, calculate the global positions of the origin of the first child's coordinate system in the hierarchy.
- Using this calculated global position, calculate the global position of the origin of the next child's coordinate system.
- Continue until you have reached the desired segment.

In the abbreviated example shown in [Figure G-4](#), only the first and last four frames are shown for the first three segments. In the actual file, all 196 frames for each of the 20 segments would appear. After all the segments, an **[EndOfFile]** section terminates the file.

Example of a HTR2 File

The example of a HTR2 file shown in [Figure G-5 on page G-14](#) was generated using Motion Analysis **SkB** and therefore does not have fixed bone lengths.

The information in the **[Header]**, **[SegmentName&Hierarchy]**, and **[BasePosition]** is very similar to the first example. However, note that the **FileVersion** is **2** instead of **1**. Also, notice that the use of scale factor and bone lengths are reversed from the usage in version 1 files. The bone lengths given in the base position are all 1.0. The actual bone length for each segment in each frame is given in the data section.

The data section starts at frame 1. The segment data is then given in the order defined in the **[SegmentNames&Hierarchy]** section. Segment 0 is the X, Y, and Z translation values for the root segment. The four values for each remaining segment are the X, Y, and Z rotations in degrees and the bone length in calibration units.

There is no end of file section, since the number of frames is already defined under **NumFrames** in the **[Header]** section.

Figure G-4. An Example of a HTR (Version 1) File

```

[Header]                                # Header keywords are followed by a single value
FileType htr                            # single word string
DataType HTRS                           # Hierarchical translations followed by rotations and Scale
FileVersion 1                           # integer
NumSegments 20                          # integer
NumFrames 196                           # integer
DataFrameRate 30                         # integer
EulerRotationOrder XYZ                  # one word string
CalibrationUnits mm                     # one word string
RotationUnits Degrees                   # one word string
GlobalAxisofGravity Y                   # character, X or Y or Z
BoneLengthAxis Y
ScaleFactor 1.0000

[SegmentNames&Hierarchy]
#CHILD      PARENT
LowerTorso   GLOBAL
UpperTorso   LowerTorso
LCollarBone  UpperTorso
RCollarBone  UpperTorso
LUpArm       LCollarBone
RUpArm       RCollarBone
LLowArm      LUpArm
RLowArm      RUpArm
LHand        LLowArm
RHand        RLowArm
LPelvis      LowerTorso
RPelvis      LowerTorso
LThigh       LPelvis
RThigh       RPelvis
LLowLeg      LThigh
RLowLeg      RThigh
LFoot        LLowLeg
RFoot        RLowLeg
Neck         UpperTorso
Head         Neck

[BasePosition]
#SegmentName Tx, Ty, Tz, Rx, Ry, Rz, BoneLength
LowerTorso238.320832923.726971241.2948288.8069650.0000002.422863 141.720766
UpperTorso0.000000141.7207660.000000-9.396187-0.112582-0.4.226674324.970754
LCollarBone0.000000324.9707540.000000-8.9120410.520925-117.992062155.689602
RCollarBone0.000000324.9707540.000000-7.516124-0.516778118.838556127.553756
LUpArm 0.000000155.6896020.000000 11.284636 -3.556895 24.261557 273.483757
RUpArm 0.000000127.5537560.000000 12.990029 2.960321 -24.250841 285.322188
LLowArm0.000000273.4837570.000000 -15.450962 -0.327390 -5.501311 318.246332
RLowArm0.000000285.3221880.000000 -12.465152 -0.572074 -5.115813 305.910223
LHand 0.000000318.2463320.000000 10.682556 -5.620492 27.438027 85.440039
RHand 0.000000305.9102230.000000 5.730878 2.097665 -20.038204 98.351413
LPelvis0.0000000.0000000.000000 24.672467 -6.972213 -127.877478122.309825
RPelvis0.0000000.0000000.000000 39.808415 6.985094 127.739824 133.924520
LThigh 0.000000122.3098250.000000 -12.585726 -15.440180-52.153190388.012318

```

#Beginning of Data. Separated by tabs

[LowerTorso]

#Fr	Tx	Ty	Tz	Rx	Ry	Rx	SF
1	1262.497925	-15.182068	-2245.441895	0.533624	0.713565	-1.069765	1.000000
2	1262.534546	-15.109411	-2245.428223	0.590220	0.697084	-1.108888	1.000000
3	1262.367920	-15.051557	-2245.315186	0.588248	0.707301	-1.163196	1.000000
4	1261.811279	-15.027791	-2245.133545	0.606978	0.696131	-1.326122	1.000000
.
.
.
193	-25.763012	-11.505680	-116.749229	-85.786575	82.456894	-104.460732	1.000000
194	-31.705627	-12.149048	-131.977005	-81.803299	80.946663	-100.766319	1.000000
195	-40.086079	-12.445135	-144.204391	-77.857658	79.605011	-97.439949	1.000000
196	-49.986629	-12.544500	-152.518951	-74.274492	78.596512	-94.241167	1.000000

[UpperTorso]

#Fr	Tx	Ty	Tz	Rx	Ry	Rx	SF
1	0.000000	0.000000	0.000000	-2.259649	-2.946399	-0.475843	1.000000
2	0.000000	0.000000	0.000000	-2.341502	-2.767608	-0.420984	1.000000
3	0.000000	0.000000	0.000000	-2.331526	-2.552349	-0.414928	1.000000
4	0.000000	0.000000	0.000000	-2.328715	-2.410058	-0.305621	1.000000
.
.
.
193	0.000000	0.000000	0.000000	2.688713	-0.354111	4.239368	1.000000
194	0.000000	0.000000	0.000000	0.312819	-2.503460	6.899297	1.000000
195	0.000000	0.000000	0.000000	-1.617305	-3.427358	9.416446	1.000000
196	0.000000	0.000000	0.000000	-2.357207	-3.057341	10.941742	1.000000

[LCollarBone]

#Fr	Tx	Ty	Tz	Rx	Ry	Rx	SF
1	0.000000	0.000000	0.000000	2.648904	1.239630	0.547259	1.000000
2	0.000000	0.000000	0.000000	2.753487	1.291442	0.549240	1.000000
3	0.000000	0.000000	0.000000	2.832925	1.339503	0.493109	1.000000
4	0.000000	0.000000	0.000000	2.839463	1.330919	0.571445	1.000000
.
.
.
193	0.000000	0.000000	0.000000	2.837519	0.820427	3.997483	1.000000

194	0.0000000	0.0000000	0.0000000	1.948793	0.402037	3.829116	1.000000
195	0.0000000	0.0000000	0.0000000	1.872437	0.383394	3.696403	1.000000
196	0.0000000	0.0000000	0.0000000	2.747322	0.820163	3.692526	1.000000
.							
.							
.							
.							
[EndOfFile]							

Figure G-5. An Example of a HTR2 File

```

Hierarchical Translation and Rotation (.htr) file
# Generated by Cortex
[Header# Header keywords are followed by a single value
FileType htr# single word string
DataType HTRS# Hierarchical translations followed by rotations and Scale
FileVersion 2# integer
NumSegments 20# integer
NumFrames 511 # integer
DataFrameRate 60# integer
EulerRotationOrder ZYX# one word string
CalibrationUnits mm# one word string
RotationUnits Degrees# one word string
GlobalAxisofGravity Y# character, X or Y or Z
BoneLengthAxis Y
ScaleFactor 1

[SegmentNames&Hierarchy]
#CHILDPARENT
HeadNeck
NeckUpperTorso
UpperTorsoLowerTorso
LCollarBoneUpperTorso
RCollarBoneUpperTorso
LUpArmLCollarBone
RUpArmRCollarBone
LLowArmLUpArm
RLowArmRUpArm
LHandLLowArm
RHandRLowArm
LowerTorsoGLOBAL
LPelvisLowerTorso
RPelvisLowerTorso
LThighLPelvis
RThighRPelvis
LLowLegLThigh
RLowLegRThigh
LFootLLowLeg
RFootRLowLeg
[BasePosition]
#SegmentNameTxTyTz Rx Ry RzBoneLength
Head 0.0 1.0 0.0 0.0 0.0 0.0 1.0
Neck 0.0 1.0 0.0 0.0 0.0 0.0 1.0
UpperTorso 0.0 1.0 0.0 0.0 0.0 0.0 1.0
LCollarBone 0.0 1.0 0.0 0.0 0.0 0.0 1.0
RCollarBone 0.0 1.0 0.0 0.0 0.0 0.0 1.0
LUpArm 0.0 1.0 0.0 0.0 0.0 0.0 1.0
RUpArm 0.0 1.0 0.0 0.0 0.0 0.0 1.0
LLowArm 0.0 1.0 0.0 0.0 0.0 0.0 1.0
RLowArm 0.0 1.0 0.0 0.0 0.0 0.0 1.0
LHand 0.0 1.0 0.0 0.0 0.0 0.0 1.0
RHand 0.0 1.0 0.0 0.0 0.0 0.0 1.0
LowerTorso 0.0 0.0 0.0 0.0 0.0 0.0 1.0

LPelvis 0.0 0.0 0.0 0.0 0.0 0.0 1.0

```

ANC—Analog ASCII Row Column

ANC (**.anc**) files contain ASCII analog data in row-column format. The data is derived from **.anb** analog binary files. These binary **.anb** files are generated simultaneously with video **.vc** files if an optional analog input board is used in conjunction with video data capture.

To create an **.anc** file from an **.anb** file, from the main menu select **File > Export ANC**. The data in ANC files is raw analog data in ASCII form and can be read and manipulated by a spreadsheet program.

Shown in [Figure G-6](#) is the beginning portion of an **.anc** file.

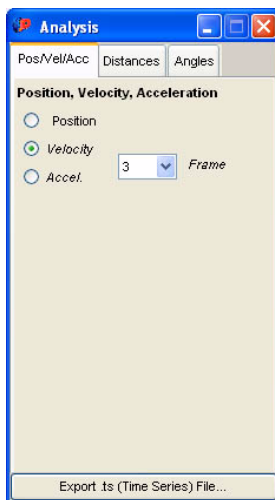
Figure G-6. Example of an ANC File

File_Type: Analog R/C ASCII					Generation#:1									
Board_Type: National AT-MIO-64F-5					Polarity:Bipolar									
Trial_Name:1ndbfw Trial#: 8					Duration(Sec.): 6.000000#Channels:30									
Name	flx	fly	flz	mlx	mly	mlz	f2x	f2y	f2z	m2x	m2y	m2z	L	tibialis
ant														
Rate	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Range	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	2500
0.0000	0	1	-29	-1	-2	0	-1	0	-13	2	3	-11	-550	
0.0010	-1	0	-29	-1	-2	0	-2	-1	-13	1	2	-10	-392	
0.0020	-1	0	-29	-1	-3	0	-2	-1	-13	1	2	-10	-369	
0.0030	-1	0	-29	-2	-3	0	-1	-1	-13	1	2	-10	-440	
0.0040	0	0	-29	-1	-2	0	-2	-1	-13	1	3	-10	-342	
0.0050	1	1	-29	-1	-2	-1	-2	0	-13	1	3	-11	-531	
0.0060	0	0	-29	-1	-2	0	-2	-1	-13	1	3	-11	-803	
0.0070	0	0	-28	-1	-2	0	-2	-1	-12	1	3	-10	-738	
0.0080	0	0	-28	-1	-2	0	-2	-1	-13	1	2	-10	-485	
0.0090	0	1	-28	-1	-2	1	0	0	-12	2	3	-10	453	
0.0100	0	0	-28	-1	-2	0	-1	0	-12	2	4	-10	876	
0.0110	0	0	-29	0	-2	0	-1	-1	-13	1	3	-10	1401	
0.0120	0	1	-29	-1	-2	0	-1	-1	-13	1	2	-10	598	
0.0130	-2	-1	-30	-3	-3	0	-3	-3	-13	0	2	-10	-141	
0.0140	0	0	-29	-1	-2	0	-2	-1	-13	2	2	-11	-457	
0.0150	0	0	-28	-1	-2	0	-2	-1	-13	2	3	-10	446	
0.0160	0	0	-29	-1	-2	0	-1	-1	-13	1	2	-11	569	
0.0170	0	0	-28	-1	-2	0	-1	-1	-13	1	3	-10	507	
0.0180	-1	0	-29	-1	-2	0	-2	-1	-13	2	3	-10	689	
0.0190	0	0	-29	-2	-2	0	-2	-2	-12	1	2	-10	501	
0.0200	-1	0	-29	-1	-2	0	-2	-1	-13	1	2	-10	348	
0.0210	-1	0	-29	0	-2	0	-1	-1	-13	1	3	-11	209	

TS—Time Series Files from the Cortex Analysis Functions: Velocity and Acceleration Calculations

Cortex Analysis Functions (F7) is a selectable view from the Data Views menu and allows the user to calculate velocity and accelerations of marker data, distances between markers and Included Angles. The Distances and Included angles are assumed to be self-documenting and a description of the Position, Velocity and Acceleration tabs is below. These data can be Exported to the .ts (Time Series) files.

Figure G-7. Analysis Functions



There is a Frames Factor, which can be set to 3, 5, 7, or 9 frames. This selects the number of frames to use for the velocity and acceleration calculations. If the Frames Factor is set to 3 frames, velocity data for the 5-th frame is calculated exclusively from frame 4 and frame 6. Velocity Data for frame 1 does not exist; velocity and frame data starts at frame 2. If the Frame Factor is set to 5, the velocity data comes exclusively from -2 frames to +2 frames from the i-th frame. Larger Frame Factors have the effect of smoothing the data.

X1, Y1, Z1 Positional data is determined from the marker locations.

Velocity Calculation is done with a central difference. Let FR represent the Frame Rate of the camera. Time difference between frames = $1 / \text{FR}$ in the below calculations.

Velocity Calculation for frame i with a Frames Factor of 3:

$$\begin{aligned} vX1(i) &= \text{FR} * (X(i+1) - X(i-1)) / 2 \\ vY1(i) &= \text{FR} * (Y(i+1) - Y(i-1)) / 2 \\ vZ1(i) &= \text{FR} * (Z(i+1) - Z(i-1)) / 2 \end{aligned}$$

Velocity Calculation for frame i with a Frames Factor of 5:

$$\begin{aligned} vX1(i) &= \text{FR} * (X(i+2) - X(i-2)) / 4 \\ vY1(i) &= \text{FR} * (Y(i+2) - Y(i-2)) / 4 \\ vZ1(i) &= \text{FR} * (Z(i+2) - Z(i-2)) / 4 \end{aligned}$$

Velocity Calculation for frame i with a Frames Factor of 7:

$$\begin{aligned}vX1(i) &= FR * (X(i+3) - X(i-3)) / 6 \\vY1(i) &= FR * (Y(i+3) - Y(i-3)) / 6 \\vZ1(i) &= FR * (Z(i+3) - Z(i-3)) / 6\end{aligned}$$

Velocity Calculation for frame i with a Frames Factor of 9:

$$\begin{aligned}vX1(i) &= FR * (X(i+4) - X(i-4)) / 8 \\vY1(i) &= FR * (Y(i+4) - Y(i-4)) / 8 \\vZ1(i) &= FR * (Z(i+4) - Z(i-4)) / 8\end{aligned}$$

Resultant velocity scalar:

$$vR1(\text{frame } i) = \text{SQRT}(vX1^{**2} + vY1^{**2} + vZ1^{**2})$$

Accelerations for Frame i are calculated as the differences in velocity as:

$$A(\text{frame } i) = \text{Velocity}(\text{frame } i+1) - \text{Velocity}(\text{frame } i-1)$$

$$\text{Time difference between frames} = 1 / FR$$

Acceleration Calculations using the Frame Rate (FR) of the camera for a Frames Factor of 3:

$$\begin{aligned}aX1(i) &= FR * FR * (X(i+1) - 2 * X(i) + X(i-1)) \\aY1(i) &= FR * FR * (Y(i+1) - 2 * Y(i) + Y(i-1)) \\aZ1(i) &= FR * FR * (Z(i+1) - 2 * Z(i) + Z(i-1))\end{aligned}$$

For Frames Factor of 5:

$$\begin{aligned}aX1(i) &= FR * FR * (X(i+2) - 2 * X(i) + X(i-2)) / 4 \\aY1(i) &= FR * FR * (Y(i+2) - 2 * Y(i) + Y(i-2)) / 4 \\aZ1(i) &= FR * FR * (Z(i+2) - 2 * Z(i) + Z(i-2)) / 4\end{aligned}$$

For Frames Factor of 7:

$$\begin{aligned}aX1(i) &= FR * FR * (X(i+3) - 2 * X(i) + X(i-3)) / 9 \\aY1(i) &= FR * FR * (Y(i+3) - 2 * Y(i) + Y(i-3)) / 9 \\aZ1(i) &= FR * FR * (Z(i+3) - 2 * Z(i) + Z(i-3)) / 9\end{aligned}$$

For Frames Factor of 9:

$$\begin{aligned}aX1(i) &= FR * FR * (X(i+4) - 2 * X(i) + X(i-4)) / 16 \\aY1(i) &= FR * FR * (Y(i+4) - 2 * Y(i) + Y(i-4)) / 16 \\aZ1(i) &= FR * FR * (Z(i+4) - 2 * Z(i) + Z(i-4)) / 16\end{aligned}$$

Resultant acceleration scalar:

$$aR1(\text{frame } i) = \text{SQRT}(aX1^{**2} + aY1^{**2} + aZ1^{**2})$$

Binary Files—ANB, TRB, and C3D

The following are binary files and cannot be directly read or manipulated by the end user. Their function and context are briefly described.

ANB

These files contain up to 64 channels of analog data collected simultaneously with video data by the optional analog board. The data in these files can be converted to readable ASCII form as either an ANA or ANC file. To do so, from the main menu select **File > Export ANC**.

Note: The data contained in the *.**anb** file has a dynamic range of –2048 to +2047, which represents 12 bit signed numbers. **Cortex** scales the specified input voltage range to this range of values.

TRB

These files contain the same 3D track data as ASCII **.trc** files, saved in a compact binary form. In addition to the data in **.trc** files, **.trb** files contain the following data for each frame:

- a list of the cameras used to calculate the 3D marker position
- the residual of the 3D position calculation

C3D

These are a special binary files that contains both scaled 3D track data and unscaled analog data. For more information, you can visit the C3D website at www.c3d.org.

Appendix H ***SIMM Motion Module***

Topic	Page
Introduction	H-1
Opening Tracked Marker Files	H-2
Analog Data	H-6
Using the Mocap Model	H-9
Analog Configuration Files	H-16
SIMM Motion Module Guide to Mocap Model Markers	H-17

Introduction

The Motion Module is an optional component to SIMM (Software for Interactive Musculoskeletal Modeling) that allows you to easily import data recorded by a motion capture system. It reads files containing tracked marker data (3D positions of markers in global space) using the TRC or TRB file format developed by Motion Analysis Corporation. It can also read analog files in the ANB or ANC format with ground-reaction force and EMG data that was recorded in sync with the motion. The Motion Module can also read C3D files, which contain both tracked marker and analog data in the same file. Additionally, the real-time version of the Motion Module can connect to a Motion Analysis system and receive and display motion and analog data in real-time, as it is being recorded.

Files of tracked marker data contain a sequence of frames, each representing a snapshot of the subject's motion at a particular instant in time. Each frame contains the X, Y, and Z coordinates, expressed in a global coordinate system, of all the identified markers. A frame of marker data can thus be thought of as a "marker cloud" because the coordinates are not organized by body segment.

The Motion Module imports tracked marker data and fits a SIMM model within the marker cloud for each time frame. If the SIMM model contains markers whose names and positions match those of the markers placed on the subject, the Motion Module can adjust the model's gencoord values to determine a "best fit" of the model to the marker cloud. The quality of a fit is determined by how closely each of the model's markers is to its corresponding marker in the marker cloud. It then uses this best fit as the starting position for solving the next frame of data. The result is a SIMM motion that matches the tracked marker data. The model that is used to fit the data can either be one that you create or the pre-made model (the *mocap model*) that comes with the Motion Module.

The Motion Module has two primary components. The first component reads files containing tracked marker data (in the TRC, TRB, or C3D format) and creates SIMM motions from them, as described above. For more information on how this process works, and the various options for importing marker data, reference the *SIMM User's Guide*.

The second component of the Motion Module creates a musculoskeletal model of a given individual by scaling a generic full-body model (the *mocap model*) based on tracked marker data from a static pose. The algorithms that are used to scale the model are the same as those used in OrthoTrak, a full-body gait analysis package available from Motion Analysis. For more information on the mocap model and how it is created and used, see Section 5.3, Using the Mocap Model.

Opening Tracked Marker Files

SIMM can import tracked marker data that is stored in either a TRB or TRC data file. These file formats, described in the **EVa** and **Cortex** manuals, contain X, Y, and Z coordinates for each identified marker for each time frame. You can also import analog data files containing forceplate and EMG data recorded during the motion. These analog data files can be in either the ANB or ANC formats. The Motion Module can also read XLS files containing other motion-related data that you may want to view in SIMM, such as the kinetic data contained in an OrthoTrak single trial spreadsheet. For more information on importing analog and XLS files, see Section 5.2.3, Analog Data.

The Motion Module can also read C3D data files. These files contain tracked marker and analog data in the same file, so you only need to load one file to import all of your motion data from a trial.

When you open a tracked marker file (along with any associated analog files), SIMM attempts to map the data onto the current musculoskeletal model, thus creating a SIMM motion that is linked to the model. Therefore, to open a tracked marker file, you must already have loaded into SIMM a model that contains the same marker set used in the marker file. For best results, you should make sure that every marker in the tracked marker file is also in the SIMM model, and that their locations in the SIMM model match where they were placed on the subject. The marker names should match exactly (except that they are case-insensitive). If the file contains markers that are not in the model, their data will be ignored by the Motion Module. Similarly, if the model contains markers that are not in the file, they will not be used to help fit the model to the motion data.

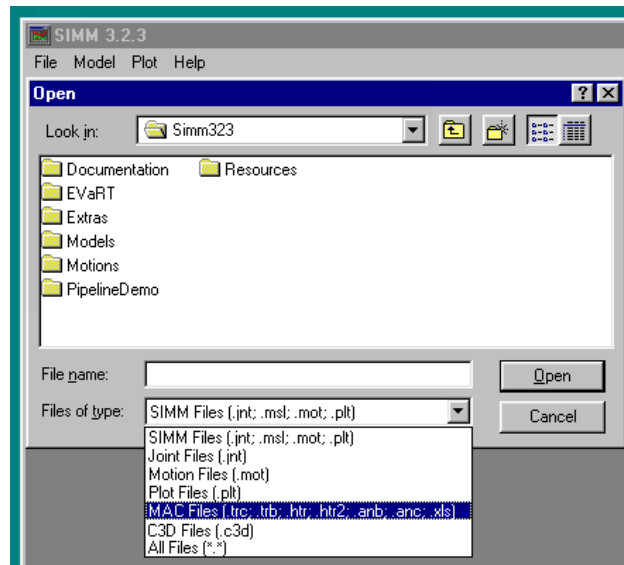
If you need to add, rename, or move markers in your SIMM model before loading a tracked marker file, you can use the Marker Editor to do so. See Section 2.12, Marker Editor, for more details.

Selecting Tracked Marker Files

To import a tracked marker file into SIMM, first make sure that the model you want to apply it to is the current model (the topmost window in SIMM). Then select **File > Open** from the menu bar. When the Windows[®] file browser appears, change the Files of Type popup menu at the

bottom of the browser to **MAC Files** or to **C3D Files** then navigate to the folder containing the tracked marker file[s] you want to import.

Figure H-1. Windows File Browser



Next, select the appropriate marker file[s] in the file browser. Click the **Open** button to import the file[s]. At this point SIMM will display a dialog box allowing you to specify several options for importing each data file into SIMM.

Note: If your analog data files have the same base name as your TRB/TRC file (e.g., **subject14.trc**, **subject14.anb**, and **subject14.xls**), then it is not necessary to select analog files in the file browser. SIMM will automatically open any analog or XLS files with the same base name and in the same folder as the tracked marker file (there is an option in the dialog box to turn off this feature). If you are loading C3D files, this is not an issue since all of the data for the motion are stored in the C3D file.

Tracked Marker Options Dialog

Once you have selected one or more tracked marker files using the process described in the previous section, SIMM displays a dialog box for each one (in sequence), allowing you to set some options for importing the marker data. In many cases, you will want to use the default settings for these options, so you can simply click the **OK** button to import the motion. The following list describes each option in the dialog box.

Import Frames, To, Increment

These fields allow you to specify the range of frames to read from the marker file, as well as the increment. To use them, type into the first two fields the starting and ending frame numbers that you want to import. The third field specifies the increment to use when reading frames from the file. For example, to read every other frame from the file, enter an increment of 2. The starting frame number field and the increment field are initialized to 1. The ending frame number is initialized to the number of frames in the marker file.

Quick Solve

The Motion Module contains two optimization algorithms for fitting the musculoskeletal model to the marker data. The default method is fairly robust- it is designed to handle cases in which several markers are missing from a frame or in which the markers move large amounts between frames. The other method, called quick solve, is less robust but works up to twice as fast as the default method. If speed is an issue, and you know that your marker data is well-behaved, you may want to turn this option on to use the faster optimization algorithm.

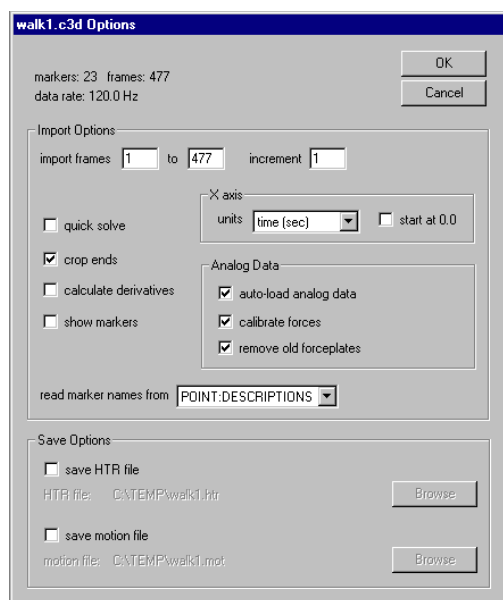
Crop Ends

Tracked marker data files often have frames at the beginning and end of a motion that are missing some markers (because the subject is outside the camera volume). To automatically detect and ignore these frames as the file is read, turn on this option (it is on by default). When the option is on, SIMM will start at the first frame and delete it if it is missing one or more markers. It will then continue to scan forward through the frames, deleting each one, until it encounters a frame containing all of the markers. It will then do the same procedure starting at the last frame and working backwards. SIMM will not remove frames with missing markers that are in between full frames, so there may still be frames in the motion that are missing markers.

Calculate Derivatives

When loading a motion, SIMM has the capability of calculating derivatives of the motion variables. When this option is turned on, after SIMM has solved the marker data and created a SIMM motion, it will calculate first-order time derivatives of the generalized coordinate values (*i.e.*, joint velocities) during the motion. It will also calculate derivatives of any force or EMG data in the analog file (if present). These derivatives can then be plotted using the Motion Curves command in the Plot Maker (see Section 2.5.2 for more details).

Figure H-2. Tracked Marker Import Dialog Box



Show Markers

This checkbox turns on the display of the global marker positions in each frame when playing back a motion. When it is on, SIMM will add spherical motion objects to the motion, representing the location of each marker, as recorded in the marker file. When you animate the model according to the motion, the blue spheres represent these actual, recorded marker locations. These are the marker locations that the Motion Module is trying to fit the model to, for each frame. It can be helpful to display them in the model window in order to visualize how good the fit is, and to help debug problems with the data.

X Axis Units, Start at Zero

These options give you control over the specification of the X axis of the motion that is created from the marker data. The units along the X axis can be either time (in seconds), or frame number. The starting X value of the motion will be 0.0 if the units are time, and 1 if the units are frame number, unless frames of data are cropped because of missing markers (see [“Crop Ends” on page H-4](#)). For example, if 12 frames of data are cropped from the beginning of the motion (and the data frequency is 60 Hz), the starting X value will be 0.2 seconds for units of time, and 13 for units of frame number. If you want the X values to start at 0.0 (or 1 for frame number) even if frames are cropped, turn on the start at zero option.

Auto-Load Analog Data

When this box is checked, SIMM will look for and automatically load any analog (ANB, ANC) or XLS data files with the same base name as the TRB/TRC file. If SIMM did not detect the presence of any analog files when the TRB/TRC file was selected, this option is grayed out.

If you selected a C3D file with the file browser, then this box controls whether or not the analog data will be read from the C3D file.

Calibrate Forces

If an analog file is present, and the auto-load analog data box is checked (see above), then this box is active and gives you control over the calibration of the forceplate data. When this box is checked, SIMM determines the baseline of each forceplate channel and automatically subtracts these baseline values from the data, thus “zero-ing out” the force data.

Remove Old Forceplates

In order to display forceplate data that is in the analog file, SIMM creates graphical objects in the model window representing the forceplates. Each time you load a tracked marker file with corresponding analog data, SIMM creates a new graphical object for each forceplate in the file. In most cases you will want to remove the existing forceplate objects from the model when loading a new file, so that the display is not cluttered with multiple (or redundant) sets of objects. Thus this option is turned on by default. If you load a series of marker and analog files that all have the same forceplate definitions, then you should leave this option turned on.

Read Marker Names From

For C3D import only: This option allows you to choose from which parameter field in the C3D file to read the names of the tracked markers. Because the POINT:LABELS field in a C3D file is limited to four characters, some software packages (*e.g.*, Cortex) store the full marker name in the POINT:DESCRIPTIONS field. Since the marker names in the tracked file must exactly match the names used in the mocap model, if your C3D file does not contain full marker names in the POINT:DESCRIPTIONS field, you may have to edit the mocap model so that the marker names match the four-character names stored in the POINT:LABELS field.

Save HTR File

This option allows you to save an HTR file containing the motion that SIMM calculates from the marker data. This HTR file cannot be read back into SIMM, but is useful if you want to import the motion into another software package. If this box is checked, a browse button is enabled that allows you to specify the name and location of the HTR file.

Save Motion File

This option allows you to save a SIMM motion file containing the motion that SIMM calculates from the marker data. This file contains exactly the same data that is in the motion that SIMM loads onto the model. You can load this motion file into SIMM at a later time, rather than re-importing the marker file. If this box is checked, a browse button is enabled that allows you to specify the name and location of the motion file.

Analog Data

Analog data files contain forceplate and EMG data that was collected in sync with motion data. When loading C3D files, there are no separate analog files; all of the analog data is contained in the C3D file. When loading TRB/TRC files, you can load analog files only if they correspond to the chosen TRB/TRC files. If the analog file has the same base name as the TRB/TRC file, then the Motion Module will load it automatically when you select the marker file. Otherwise, you should select the analog file as well in the file browser. The same holds for XLS files, which are not actually analog files, but are treated similarly. XLS files can contain other data corresponding to the recorded motion, such as kinetic data calculated by OrthoTrak and stored in a “single trial spreadsheet.”

SIMM can recognize three types of analog data: ground reaction forces, EMG activation levels, and “other” data (usually kinetic data from an XLS file). These data types, and how they are interpreted by SIMM, are described below:

Forceplate Data

SIMM displays forceplate data by drawing a vector in the model window at the appropriate point of application and with a size corresponding to the magnitude of the force. Forceplate data in an analog file are voltages measured by forceplate transducers. These voltages are converted into forces using a calibration file, **forcepla.cal**. This file is the same one used by EVa, Cortex, and OrthoTrak. To use it with SIMM, you should put a copy of it in the same folder as your motion data, or in the folder **SIMMResources\mocap\misc**. If you have only one forceplate configuration for your motion capture system, it is preferable to put *forcepla.cal* in **SIMMResources\mocap\misc**, rather than copying it into every folder of motion data.

Note: C3D files that contain force plate data also contain the calibration information for the plates. Thus there is no separate calibration file that SIMM reads when importing C3D files.

SIMM also uses another configuration file, **importVariables.txt**, to map forceplate channels to SIMM variables. This file is located in **SIMMResources\mocap\misc**, and contains mappings for typical channel names for up to six forceplates. You will only need to change this file if you use more than six forceplates, or use forceplates that have exotic channel con-

figurations. This file is used when loading ANB/ANC files and when loading C3D files.

EMG Data

SIMM displays EMG data by varying the sizes and colors of the corresponding muscles in the SIMM model. EMG data in an analog file are voltages measured by the EMG system. SIMM rectifies and smooths these data, and then scales them based on an MVC value (maximum voluntary contraction), resulting in a smooth muscle excitation level that varies between 0.0 and 1.0. If MVC values are located in the configuration file **importVariables.txt**, SIMM will use them to scale the EMG data. If MVC values are not present, SIMM will use each muscle's maximum voltage in the analog file to scale that muscle's EMG data (thus each muscle's excitation will peak at 1.0 sometime during the motion). The file **importVariables.txt**, located in **SIMM\Resources\mocap\misc**, contains mappings between typical EMG channel names and the muscle names in the mocap model. It does not contain any MVC values. In most cases, however, it is sufficient to not specify them and use SIMM's default scaling method.

Other Data

"Other" data is contained in XLS files, and can represent any motion variable that you choose to calculate and store in the file. It is usually reserved for kinetic data (*e.g.*, joint moments and powers) that OrthoTrak calculates and stores in its spreadsheet (XLS) format. It may also include motion events, such as toe-off and heelstrike, that are stored at the top of the XLS file. SIMM does not perform any calculations on these data, but does import them so that you can create plots of them in the Plot Maker. SIMM will only import "other" data that are identified as such in **importVariables.txt**. This configuration file, located in **SIMM\Resources\mocap\misc**, contains mappings between OrthoTrak and the mocap model of all forces, moments, and powers for the hip, knee, and ankle joints. You will only need to edit this file if you want to import data other than these.

Real-time Import

In addition to importing tracked marker files, SIMM can import motion data that is sent over the network in real-time from Cortex. SIMM is thus able to animate a musculoskeletal model and plots of joint angles and muscle lengths while the subject's motion is being recorded. For this real-time connection, **Cortex** solves tracked marker data using the mocap model. It then sends generalized coordinate values (as well as analog data) over the network to the SIMM computer. If the same mocap model is loaded into SIMM, these generalized coordinates will drive the animation of the model in real-time, with a small delay (whose length depends on the network speed and the graphics speed of the SIMM computer).

Follow these steps to use the real-time connection between Cortex and SIMM:

First-time setup only:

1. Find the folder **SIMM\Cortex** on your SIMM computer and look for **mocap.jnt** and **solver.dll** (**solver.dll** may be hidden in the folder view because it is a system file). Copy both files to the folder on the **Cortex** machine (you'll need to exit **Cortex** first, if it is running). This will guarantee that **Cortex** is using the same mocap model and the same scaling algorithms as SIMM uses.

Each motion capture session:

2. Open the text file **SIMM\Resources\preferences** in a text editor such as Notepad or Wordpad. Locate the line that reads: `Cortex_MACHINE <hostname>`, and change the hostname to the name of your **Cortex** machine. Save and close the preferences file (make sure that the file is saved as ASCII text with the name preferences—Wordpad likes to surreptitiously append a **.txt** extension when it saves files that don't already have a filename extension).
3. If your motion capture system includes forceplates, copy the file **forcepla.cal** from your **Cortex** computer onto your SIMM computer and put it in the folder **SIMM\Resources\mocap\misc**.
4. Copy the folder containing the motion data from the Cortex computer to the SIMM computer (or make it shared). If there is a **personal.dat** file for this data, make sure it is in the folder too.
5. Launch **Cortex**. Load the appropriate project.
6. Select **File > Load Tracks File** and select the tracked marker file corresponding to the static trial for the subject.
7. Under **System > Misc**, click on the radio button for **SIMM OrthoTrak Solver**, located in the Skeleton Options area.
8. Launch SIMM.
9. Select **File > Open Mocap Model** and navigate to the motion folder. Choose the tracked marker file containing the static pose.
10. Set the options as desired in the dialog box and click **OK**.
11. Open the Model Viewer window.
12. In the Model Viewer window, choose **Start > Realtime Connection to <hostname>**. SIMM will display a dialog box allowing you to set some options for the connection. The motion buffer size options control how many seconds of motion data are saved in SIMM's buffer. The *time scale* options let you specify the minimum and maximum values, in seconds, for the time scale of the motion. If you want the scale to remain fixed between minimum and maximum, check the sliding checkbox, otherwise the scale will continue to increase as new data is received.
13. SIMM will now wait to receive data from the **Cortex** computer. Once the connection is established, SIMM will display "connected" in its message window, and the SIMM model will begin tracking the motion of the subject in real-time. You can pan, zoom, rotate, and change the draw mode of the SIMM model as it is tracking the motion. You can also create plots of kinematic variables and muscle properties and see the plots change in real-time.
14. To disconnect SIMM from the real-time stream, click the **Stop** button in the Model Viewer. You can play back the last N seconds of the motion.

Note: When analog data is imported into SIMM in real-time, it is processed slightly differently than when the data is post-processed in SIMM. This is because the real-time analog data is processed frame-by-frame, without the benefit of the full data set. This has the following implications:

- To set the baseline for the forceplates, the first frame of force data is used as the "zero" level. Thus when you first connect SIMM to **Cortex**, you should make sure that nothing is on the forceplates.

- If no MVC levels have been specified for some of the muscles, a running tally of each muscle's maximum level will be kept, and used to scale the EMG signals into the range 0.0 to 1.0. Thus if you want to accurately scale EMG levels throughout a real-time SIMM motion import, you should either specify MVC levels, or have the subject perform MVCs just after connecting SIMM to Cortex.

Using the Mocap Model

SIMM has the ability to read tracked marker data and convert it into a motion by fitting a musculoskeletal model to it. For this to work well, the body segment lengths, marker names, and marker locations in the model must exactly match those for the subject whose motion is being recorded. Because it is time consuming to measure and scale the body segments, and measure and record the offsets of all of the markers, the Motion Module has the ability to automatically scale a pre-made model (the mocap model) to fit the subject.

To use the mocap model, select **Open Mocap Model** from the **File** menu. SIMM will display a Windows® file browser and ask you to select the name of a static pose file. This static pose is used to calculate joint center locations and segment lengths for the subject, using the same algorithms implemented in OrthoTrak. In other words, *the Motion Module recreates the OrthoTrak skeletal model from the static pose, and then maps this skeletal model onto the mocap model*. Thus to use the mocap model, you need to use the same motion capture protocol as you would for OrthoTrak. You can use either the Helen Hayes or Cleveland Clinic marker sets (plus your own additional markers, if desired), as long as the marker names and locations match the protocol defined in the OrthoTrak manual. The Motion Module uses the tracked marker data from the OrthoTrak static pose, and also segment information from **personal.dat**, to scale the mocap model to the subject. The algorithms for calculating joint center locations and segment lengths have been designed to be as similar as possible to the OrthoTrak algorithms. This was done so that motion information in SIMM (e.g., joint angles, EMG levels) would match the corresponding information in OrthoTrak, and also so that you would not have to change your OrthoTrak protocol in order to use SIMM.

The mocap model and the algorithms used to scale it are described in the following sections.

The Mocap Model

The mocap model is a full-body SIMM model that has been customized for gait analysis, but can be used to import and display any type of full-body motion. The model has 41 body segments, 41 joints, 40 degrees of freedom, and 88 lower-extremity muscles. It represents an adult male, approximately 175 cm. tall, with a mass of 78 kg. The model is scaled to match the size of the motion capture subject using algorithms described in Section 5.3.4. The model's joints have been carefully constructed to represent normal joint motion as closely as possible.

To load the mocap model into SIMM, the software looks for the `MOCAP_MODEL` variable in the preferences file (**SIMM\Resources\preferences**) to get the name of the joint file that comprises the mocap model. The default setting for this variable is **SIMM\Resources**

mocap\mocap.jnt. This joint file includes the file **mocap.msl** to get the definitions of the muscles.

You may change the mocap model however you wish. For example, you can add or remove muscles from the model, or change the tendon and fiber parameters of existing muscles. You can also add degrees of freedom to the model, in order to more accurately represent a particular motion (*e.g.*, adding toe joints and gencoords to examine toe motion in greater detail). If you modify the mocap model, however, you should keep in mind two things.

First, the model has been set up to correspond to the skeletal model that OrthoTrak uses when processing gait data. The lower-extremity body segments and orientations of the reference frames closely match those in the OrthoTrak model. Also, each body segment in the mocap model is scaled to fit the subject by relating its length to the length of an OrthoTrak segment. These relations are specified in the mocap model by defining scale segments and scale factors for each body segment. If you add, delete, or modify joints or body segments in the mocap model, you should make sure that each segment still properly relates to an OrthoTrak segment.

Second, **mocap.jnt** contains several macros that are used to properly define the orientation of the floor, and to automatically remove the upper body segments if there are no upper body markers. When SIMM reads a joint file, it performs these macros but does not save them internally. Thus when it writes out a joint file, all of the macros have been removed. If you make changes to the mocap model in SIMM and then save the new model to a file, do not replace *mocap.jnt* with the new file. Instead, copy the relevant portions of the new file into *mocap.jnt* using a text editor, thus preserving the macros and comments.

The Static Pose

When you open the mocap model, SIMM prompts you for the name of a tracked marker file containing a static pose of the subject. This static pose is the same one used by OrthoTrak, and for it you can use any of the six marker sets identified by that software package: *Cleveland Clinic Lower Body*, *Cleveland Clinic Full Body*, *Cleveland Clinic Full Body with Head*, *Helen Hayes Lower Body*, *Helen Hayes Full Body*, and *Helen Hayes Full Body with Head*. It is also strongly recommended that you include the medial knee and ankle markers in the static pose, for more accurate calculation of knee and ankle joint centers. You can also supplement the OrthoTrak marker set with your own custom markers, as long as you do not move or remove any markers from the identified set. Lastly, the marker set used in the static trial must include all of the markers you plan to use for capturing motion. This is because the Motion Module calculates the locations of all markers in the mocap model based on their locations in the static trial. These are the steps you should follow when collecting the static trial:

1. Choose which of the six OrthoTrak marker sets you would like to use for capturing motion.
2. Add the medial knee and ankle markers, for better calculation of knee and ankle centers (not required, but highly recommended).
3. Add any additional markers that you would like to use (*e.g.*, extra markers on the feet, more markers on the arms). These markers must also be added to the mocap model, The Marker Set.

4. Capture the static trial using the protocol outlined in the OrthoTrak manual. The subject should have their arms either down by their sides, or straight out from their body with their thumbs facing forward.
5. Remove the medial knee and ankle markers, and any others that you do not want to use for capturing motion.

Note: If you use a marker set with no upper extremity markers, the Motion Module will remove the upper extremity from the mocap model and display only the pelvis and legs.

Once you have selected the static pose file to be used for opening the mocap model, SIMM displays a dialog box, allowing you to set some options for importing the static pose. In many cases, you will want to use the default settings for these options, so you can simply click the *OK* button to import the motion. The following list describes each option in the dialog box:

average from frame

These fields allow you to specify the starting and ending numbers for the sequence of frames that are averaged together to determine the static pose. These fields are initialized to 1 and the number of frames in the file, meaning that all frames will be averaged. If frames in the chosen sequence are missing some markers, locations for markers that are present will still be used in the average.

load personal.dat

This option gives you control over the automatic loading of **personal.dat**. When SIMM loads the static marker file, it looks for a file called **personal.dat** in the same folder. This file is identical to the one created and used by OrthoTrak. If the file is present, SIMM will automatically load it and read model parameters from it, such as foot length and hip origin offsets. It will use these parameters to determine joint center locations and segment lengths, using the same algorithms that OrthoTrak does. If there is no **personal.dat** file present in the folder, this option will be grayed out. If it is checked and you do not want to load **personal.dat**, click the box to turn it off.

read marker names from

For C3D import only: This option allows you to choose from which parameter field in the C3D file to read the names of the tracked markers. Because the `POINT:LABELS` field in a C3D file is limited to four characters, some software packages (e.g., **Cortex**) store the full marker name in the `POINT:DESCRIPTIONS` field. Since the marker names in the tracked file must exactly match the names used in the mocap model, if your C3D file does not contain full marker names in the `POINT:DESCRIPTIONS` field, you may have to edit the mocap model so that the marker names match the four-character names stored in the `POINT:LABELS` field.

subject mass

This field allows you to specify the total mass of the model after it has been scaled to fit the size of the subject. After the scaling is done, all the body segments' mass parameters are scaled up or down by a single percentage so that the total mass of the model equals the number entered into this field. This field has no effect if mass properties are not specified in the model file.

preserve mass distribution

This option gives you control over how the mass properties of the individual body segments are scaled. If this option is off, then each body segment's mass is scaled proportionally with its size. If this option is on, then each segment's mass parameters are not scaled with their change in size (i.e., the distribution of body mass specified in the model file is preserved). In either case, after the model has been scaled, all the body segments' mass parameters are scaled up or down by a single percentage so that the total mass of the model equals the number entered into the "subject mass" field.

Figure H-3. Static Trial Import Dialog Box

init1.c3d Options

Static C3D file:
 markers: 190 frames: 271
 data rate: 120.0 Hz

Mocap Model: C:\Users\Sim\Resources\mocap\mocap.jnt

Import Static Pose

average from frame 1 to frame 271 subject mass 75.000

☒ load personal.dat ☐ preserve mass distribution

read marker names from POINT:DESCRIPTIONS

Export Scaled Model

☐ save JNT file
 JNT file: C:\DOCUMENT1\peter\LOCALS1\Temp\init1.jnt Browse

☐ save MSL file
 MSL file: C:\DOCUMENT1\pet...LOCALS1\Temp\init1.msl Browse

OK Cancel

Save JNT File, Save MSL File

These options allow you to specify if SIMM will write out joint and muscle files containing the musculoskeletal model that is scaled to fit the subject. After SIMM has loaded the mocap model and scaled it based on the data in the static marker file and **personal.dat**, it will write out corresponding joint and muscle files, depending on the states of these check boxes. You may want to create these files so that you can make changes to them or to be able to re-load the model without going through the scaling process again.

Calculation of Joint Centers

Once the static pose has been loaded, the Motion Module recreates the OrthoTrak skeletal model from the marker cloud. The first step is determining the locations of the joint centers for all of the joints in the OrthoTrak model. The pelvis, hip, knee, and ankle centers are all found using the same procedure used by OrthoTrak.

The hip center is determined using percentage offsets from the pelvis markers. The Motion Module reads these offsets from **personal.dat**, as written by OrthoTrak. The default values for these offsets are taken from *Bell et al. Journal of Biomechanics, 23(6), 1990, pp. 617-21*:

posterior displacement: 22%

lateral displacement: 32%

inferior displacement: 34%

To change these values, edit the file **personal.dat**, as described in Appendix D of the OrthoTrak manual.

The knee and ankle centers are found using the medial and lateral markers. It is strongly recommended that you use medial markers for a more accurate calculation of joint centers. If you choose not to use them, you should enter knee and ankle diameter measurements into **personal.dat**. The Motion Module will use them to locate the knee and ankle centers if no medial markers are used.

The default method for determining shoulder, elbow, and wrist joint centers uses percentage offsets from the appropriate marker locations. If medial elbow and wrist markers are used in the static trial, their locations are averaged to get the joint centers, as is done with the knee and ankle. It is recommended that you use medial elbow and wrist markers in the static trial if you want an accurate representation of arm motion.

Scaling the Mocap Model

Once the locations of the OrthoTrak joint centers have been calculated from the static pose, the Motion Module determines the orientations of the OrthoTrak segment reference frames. It then can measure the lengths of the OrthoTrak segments and use them to scale the mocap model to match the size of the subject.

The reference frames for the foot, shank, thigh, pelvis, and torso are all determined using the procedure described in Appendix H of the OrthoTrak manual.

OrthoTrak does not create reference frames for the upper and lower arms, but the Motion Module does this using one of several methods. If medial elbow and wrist markers are used in the static trial, then the arm reference frames are found in the same way in which the thigh frames are found. If no medial markers are present, then the upper and lower arm reference frames are found using the line between the joint centers as the X axis, and using the same Y axis as the torso. The Z axis is then determined by crossing X and Y.

Once all of the segment reference frames have been determined, the length of each segment is calculated. For most segments, the length is simply the distance from one joint center to the next. For the foot, the Motion Module reads the length from **personal.dat**. If there are no foot

length measurements in **personal.dat**, then the foot length is assumed to be 1.4 times the distance from the heel marker to the toe marker (the toe marker is actually placed on the top of the foot just posterior to the toes).

Each body segment in the mocap model contains scaling information that tells the Motion Module how to scale it based on an OrthoTrak skeletal segment. The scaling information consists of the name of the OrthoTrak segment and X, Y, and Z reference numbers that correspond to the unscaled length of the SIMM segment. For example, the right femur in the mocap model contains the line:

```
gait_scale R_THIGH 0.3960 0.3960 0.3960
```

This tells the Motion Module that the unscaled femur is 0.3960 meters long. Once the length of the corresponding OrthoTrak segment is known (R_THIGH), the femur can be scaled accordingly. If the R_THIGH segment were 0.35 meters long, then the femur would be scaled by a factor of $0.35/0.396$. In most cases the three reference values are the same number, indicating that the segment should be scaled uniformly in X, Y, and Z. The two exceptions are the TORSO and PELVIS, which are scaled differently in two dimensions. For SIMM segments that do not map directly to an OrthoTrak segment, their scaling information is copied from the most relevant segment. For example, the right hand in the mocap model copies the scaling information from the right lower arm, so that the hand is scaled the same amount as the lower arm.

The Marker Set

The marker set in the mocap model that comes with SIMM includes every marker used in all six marker sets that OrthoTrak recognizes, plus the medial knee and ankle markers. In addition, many other markers have been added, such as medial elbow and wrist markers. For a complete list of the markers in the model, as well as information on when they should be used and where they should be placed on the subject, read the *Guide to Mocap Model Markers* document. The mocap model contains over 80 markers, which is more than the number used in most applications. When the static trial is loaded, any marker in the mocap model which is not in the static trial is removed from the model. Thus it is not a problem to have extra markers in the mocap model. In fact, you should add to the model whatever extra markers you may need for any of your motion capture applications. Then for a particular application the mocap model will have all the necessary markers, and the unused ones will automatically be removed when the model is loaded into SIMM.

To add or change markers in the mocap model, use the Marker Editor. You should be careful not to overwrite the original **mocap.jnt** file. Instead, after editing the marker set, save the model to a new file name, and copy the altered markers into **mocap.jnt**.

All of the markers in the mocap model have X, Y, and Z offsets that put them in realistic locations given the dimensions of the generic model. Thus if you load the unscaled mocap model into SIMM (using the **File > Open** command, not the **File > Open Mocap Model** command, which will scale it), the markers will appear in positions corresponding to where they are placed on the subject. *These offsets are purely decorative, to help you view the marker set. They are not used by the Motion Mod-*

ule to process any marker data. To explain why this is so, we must first introduce the concept of *critical* markers and *non-critical* markers.

Critical markers are ones that must be present in the static trial in order for the Motion Module to load and scale the mocap model. For the lower body, these markers are: V.Sacral, R.ASIS, L.ASIS, R.Knee (or R.Knee.Lateral), R.Ankle (or R.Ankle.Lateral), R.Heel, R.Toe, L.Knee (or L.Knee.Lateral), L.Ankle (or L.Ankle.Lateral), L.Heel, and L.Toe. If any of these markers is missing from the static trial, the SIMM model of the lower body will not be loaded. For the upper body, the critical markers are: V.Sacral, R.ASIS, L.ASIS, R.Shoulder, R.Elbow, R.Wrist, L.Shoulder, L.Elbow, L.Wrist. Note that the ASIS and sacral markers are critical for both portions of the body. If one of these markers is missing from the static trial, you will get an error when trying to load the mocap model. Non-critical markers are all other markers in the set.

Once the Motion Module has determined the locations of the joint centers and the orientations of the segment reference frames from the static pose, it calculates the proper offsets for all of the critical markers (plus the static-only medial markers). For example, once the right thigh reference frame has been oriented within the static pose marker cloud, the exact positions of the critical markers attached to the right thigh can be measured directly from the static pose and entered into the mocap model, thus overwriting whatever offsets were in the model input file.

After the offsets of all the critical markers have been determined in this fashion, the mocap model is “fit” to the static pose marker cloud using only the critical markers to find the best fit. This process orients the mocap model within the marker cloud, so that the offsets of the non-critical markers can be measured directly from the static pose. These offsets are then entered into the model, overwriting whatever values were in the model input file.

To summarize, the Motion Module uses a two-step process to calculate proper offsets for all of the markers in the mocap model. The first step determines the offsets of the critical markers, which the OrthoTrak algorithms can definitively locate without knowing anything about the mocap model. Then these critical markers are placed on the mocap model, and the model is fit to the static pose marker cloud. Now the offsets of the other markers can be measured, because every body segment in the mocap model is now correctly placed in the static pose.

Analog Configuration Files

SIMM can include analog data such as ground reaction forces, EMG activation levels, and kinetic data when importing a motion. SIMM uses a configuration file named **importVariables.txt** to determine which analog variables to import from an analog file, and how the data for each variable should be interpreted. This configuration file is used for both TRB/TRC import (with corresponding ANB/ANC analog files) and for C3D import (where the analog data is contained in the C3D file itself). SIMM can interpret analog data as one of three types:

Forceplate Data

These variables specify voltages representing force or moment components as measured by a forceplate transducer. Given the voltages generated by a forceplate (6 channels for an AMTI or Bertec forceplate, 8 channels for a Kistler forceplate) SIMM can calculate and display a force location and vector for the forceplate.

EMG Data

These variables define activation levels for one or more muscles in the SIMM full-body model. SIMM rectifies, smooths, and scales EMG data so that it can be plotted, and used to control the width and color of muscles during an animation.

Other Data

Any variables that are not forceplate or EMG data are classified as other data. SIMM does not perform any calculations on these data variables, but they may be included in SIMM plots.

importVariables.txt

The **importVariables.txt** file, located in **SIMM\Resources\mocap\misc**, contains a list of variable names and attributes. When SIMM processes an analog data file or an OrthoTrak XLS file, it consults the **importVariables.txt** file to decide which variables to import and how to interpret them.

Each row in **importVariables.txt** defines a variable to be imported. The first column in a row specifies the name of the variable as it appears in the analog or XLS data file. Since certain analog files support variable names with spaces in them, the first column of the **importVariables.txt** file *must* be terminated by a tab character. SIMM considers all characters from the beginning of a row until the first tab character to be the name of the import variable. SIMM does a case-insensitive comparison when matching variable names defined in **importVariables.txt** with variable names in an analog data file. Therefore the name “Rt Tibialis” would be considered the same as “rt tibialis”.

The second column in a variable definition specifies the type of the variable. It must be one of the following keywords: `force_plate`, `muscles`, or `other_data`. These keywords must be lowercase. Following each keyword is information describing the variable:

This keyword specifies a ground reaction force variable. It must be followed by three values:

1. The forceplate number (1, 2, 3, etc.), then
2. The keywords `force` or `moment`, then

3. The channel component (x, y, or z for AMTI or Bertec forceplates, or x12, x34, y14, y23, z1, z2, z3, z4 for Kistler forceplates).

This keyword specifies an EMG variable. It must be followed by one or more SIMM muscle names. The keyword `mvc` may optionally appear after the last muscle name. If `mvc` appears, then it must be followed by an integer number that SIMM uses as the voltage for the maximum voluntary contraction when scaling that EMG channel. If no MVC value is specified, then the channel is scaled such that its maximum value is 1.0. EMG scaling is performed after the EMG channel's data has been smoothed and resampled to the motion's frequency.

This keyword specifies a data channel that exists simply to be included in SIMM plots. This keyword may be optionally followed by a single word that will be used to label this channel in SIMM plots. If no name follows the `other_data` keyword, then the name of the imported variable will be used.

forcepla.cal

When importing analog data from ANB/ANC files, SIMM uses the same calibration file as Cortex and OrthoTrak for processing forceplate data. Therefore, you can simply copy the **forcepla.cal** file from your Cortex folder into the **Resources\mocap\misc** folder. For users who need to create a **forcepla.cal** file to describe their forceplate(s), refer to Appendix C of the OrthoTrak manual.

Note: **forcepla.cal** is not used for C3D import since C3D files contain the necessary calibration information for the force plates.

SIMM Motion Module Guide to Mocap Model Markers

This guide describes the markers used by the Motion Module in SIMM to load each Mocap Model, scale it to fit the subject, and import recorded motions. For details on how the Motion Module processes the marker data and the model, see Chapter 5 of the SIMM User Guide. For a tutorial of the Motion Module, click on Help-> SIMM Tutorials -> Motion Module Demo in the SIMM menu bar. This document focuses on the names and locations of the markers, and when they are needed by the Motion Module.

Definitions

Static Trial

A TRC, TRB, or C3D file of a motion capture subject in a static pose, usually the "T" or "scarecrow" pose

Motion Trial

A TRC, TRB, or C3D file of a subject performing an activity, such as walking or throwing

Mocap Model

A SIMM musculoskeletal model that can be loaded into SIMM, scaled to fit a subject using a static trial, and used to animate motion trials of that subject. The primary model is a full-body model with lower-extremity muscles, but others are available as well.

Critical Marker

A marker that is required in the static trial, and which must be placed in a specific location on the subject, according to instructions in the OrthoTrak manual. The coordinates of the marker in the static trial are used to determine joint centers and body segment lengths.

Semi-critical Marker

A marker that is optional in the static trial, but if used, must be placed in a specific location on the subject, according to instructions in the OrthoTrak manual. The coordinates of the marker in the static trial are used to improve the joint center calculations.

Optional Marker

A marker that is optional in the static trial, and whose placement on the subject does not need to be in a specific location

Fixed Marker

An optional marker whose X, Y, Z offsets are not automatically calculated when the static trial is processed. Rather, the offsets in the marker definition in the Mocap Model file are used to position the marker on the model (these offsets are scaled with the body segment, however).

The Motion Module comes with four different Mocap Models for you to choose from. Each of them contains parameters that turn on and off different portions of the model, depending on which of the critical markers are present in the static trial. When you load a Model Model with a static trial, the Motion Module reads the list of markers from the trial and sets the values of the model parameters so that the appropriate portions are included. For example, if the critical markers on the right hand are present, then the degrees of freedom in the fingers are activated. If they are not present, the hand is modeled as one rigid body segment, with movement only at the wrist.

The Mocap Model that you will most likely want to use is **mocap.jnt**. This is a model of a full body, with lower extremity muscles and [optionally] movable fingers in each hand. There is also a right arm model and a left arm model (**rightArm.jnt** and **leftArm.jnt**). These should be used if you want to capture motion of one arm without any torso or pelvis markers. Lastly, **mocap3D.jnt** is similar to **mocap.jnt**, but it includes 3D muscle surfaces for 18 key lower extremity muscles, rather than the lines of action for all 86 muscles. These muscle shapes look more realistic, but they do not have force-generating parameters, so you cannot calculate the lengths or forces in these muscles during the recorded motion.

Table H-1 on page H-19 shows the available combinations of model components. To determine which Mocap Model you should use, find the row that best describes the model you want, then locate the filename in the last column. All of these files are located in **SIMM\Resources\mocap**. Once you have determined which one to use, you can either set the MOCAP_MODEL variable in **SIMM\Resources\preferences** to that file, or choose that file using the **Options...Choose Model Model** command in the SIMM menu bar.

Table H-1. Combinations for Model Components

lower extremity	upper extremity	movable fingers	muscles	file name
yes	yes	yes	legs only	mocap.jnt
yes	yes	no	legs only	mocap.jnt
yes	no	no	legs only	mocap.jnt
no	yes	yes	none	mocap.jnt
no	yes	no	none	mocap.jnt
no	right arm only	yes	none	rightArm.jnt
no	right arm only	no	none	rightArm.jnt
no	left arm only	yes	none	leftArm.jnt
no	left arm only	no	none	leftArm.jnt
yes	yes	yes	legs only, 3D	mocap3D.jnt
yes	yes	no	legs only, 3D	mocap3D.jnt
yes	no	no	legs only, 3D	mocap3D.jnt

It is important to note that *the critical and semi-critical labels for markers are relevant only for the static trial. For motion trials, all markers are optional.* That is, after recording the static trial, you can remove any of the markers from the subject before recording motion trials. Generally, however, you will want to keep all of the markers on the subject for the motion trials, with the possible exception of the medial joint markers. Also, once the static trial has been recorded, you must be careful not to move any of the markers on the subject (except for removing them completely). SIMM uses the static trial to calculate the coordinates of each marker relative to its body segment, so if you move a marker or add additional markers, you must re-record the static trial and re-load the Mocap Model.

All of the markers described in this document are already part of the primary Mocap Model, located in *SIMM\Resources\mocap\mocap.jnt*. To use any of them, you do not need to make any changes to the file; just place the markers on the appropriate locations on the subject, and make sure the marker names in the static trial match the names shown in the figures below. Many of the markers can have one of several names, as listed

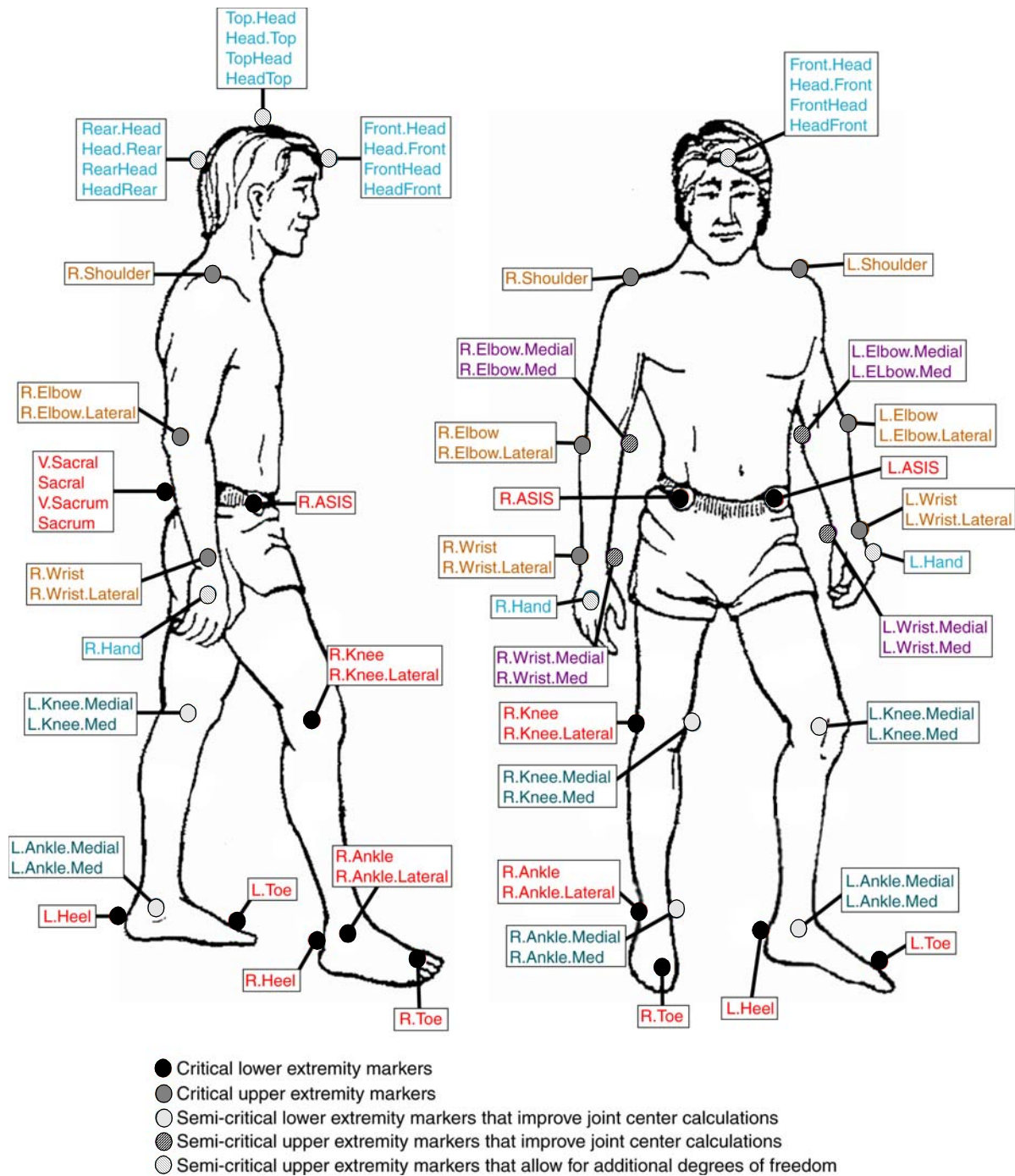
in the box pointing to each marker in the figures. These names are case-insensitive, and may contain spaces.

If you want to add markers to the Mocap Model, you can do so with the Marker Editor in SIMM. This tool allows you to create new markers, attach them to the appropriate body segments, and specify their X, Y, Z offsets. The exact values of the offsets are not important; they are used only for display of the marker while creating it. The offsets will be overwritten with values calculated by the Motion Module when the static trial is processed and the model is scaled to fit the subject. This process is described in more detail in Chapter 5 of the SIMM User Guide, but here is a brief summary. After loading the static trial, the Motion Module places all of the critical markers that are in the trial on the Mocap Model in their corresponding locations. The Mocap Model is then scaled to match the subject, and then a least-squares optimization fits the model within the cloud of static trial markers, considering only the critical markers. This positions the model within the marker cloud so that the Motion Module can then directly calculate the offsets from the optional markers to the model segments to which they are attached. If you do not want the offsets for a marker to be calculated in this manner, then you must turn on the “fixed” button for that marker in the Marker Editor, and enter accurate X, Y, Z offsets into the number fields. This tells the Motion Module to scale the marker's offsets when the model is scaled, but not to recalculate their values as it does for the other optional markers.

Note on adding markers: You can create new markers using the Marker Editor, and then save the model by writing out a joint file, but you should not replace the original model file (*e.g.*, **SIMM\Resources\mocap\mocap.jnt**) with this new file. This is because the model file contains many comments and special parameters that enable SIMM to automatically modify it for a particular static trial, as described above. However, when this file is loaded into SIMM and then written back out, these comments and parameters are lost. Thus after saving your new joint file, you should use a text editor to copy the new marker definitions from the file and paste them into the existing model file.

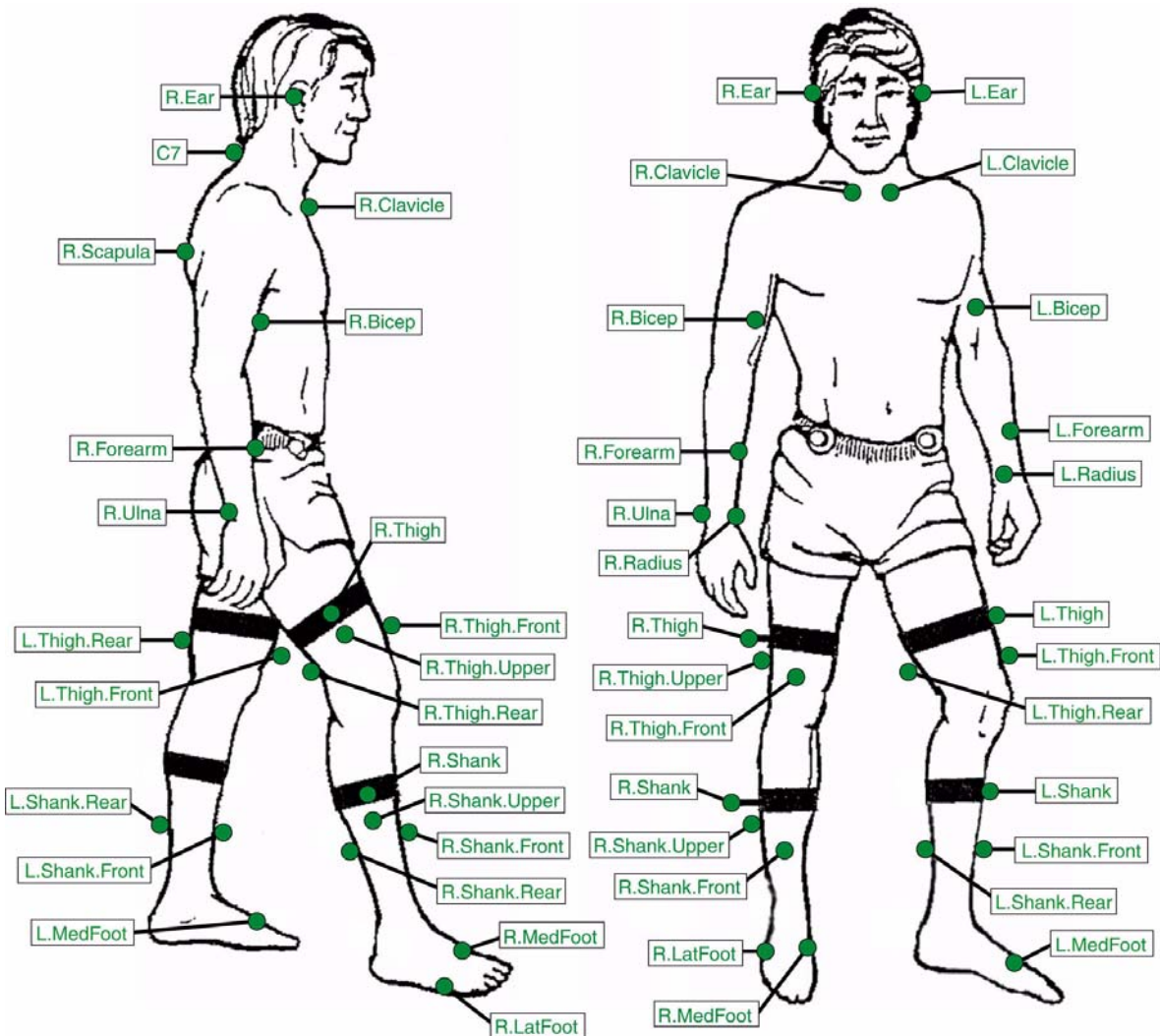
Shown in [Figure H-4 on page H-21](#) are the critical and semi-critical markers for upper body and lower body motion recording. If any of the lower body critical markers is missing from the static trial, the legs will not be loaded with the Mocap Model. Similarly, if any of the upper body critical markers is missing from the static trial, the torso, head, and arms will not be loaded. Note that the sacral, left ASIS, and right ASIS markers are critical for both upper and lower body motion recording. If any of these markers is missing, the Motion Module will print an error and not load the Mocap Model. The head and hand markers are semi-critical. If used, they allow the Motion Module to track motion at the neck and wrist. If not used, these joints will remain fixed during animation of motion trials in SIMM.

Figure H-4. Critical and Semi-Critical Markers



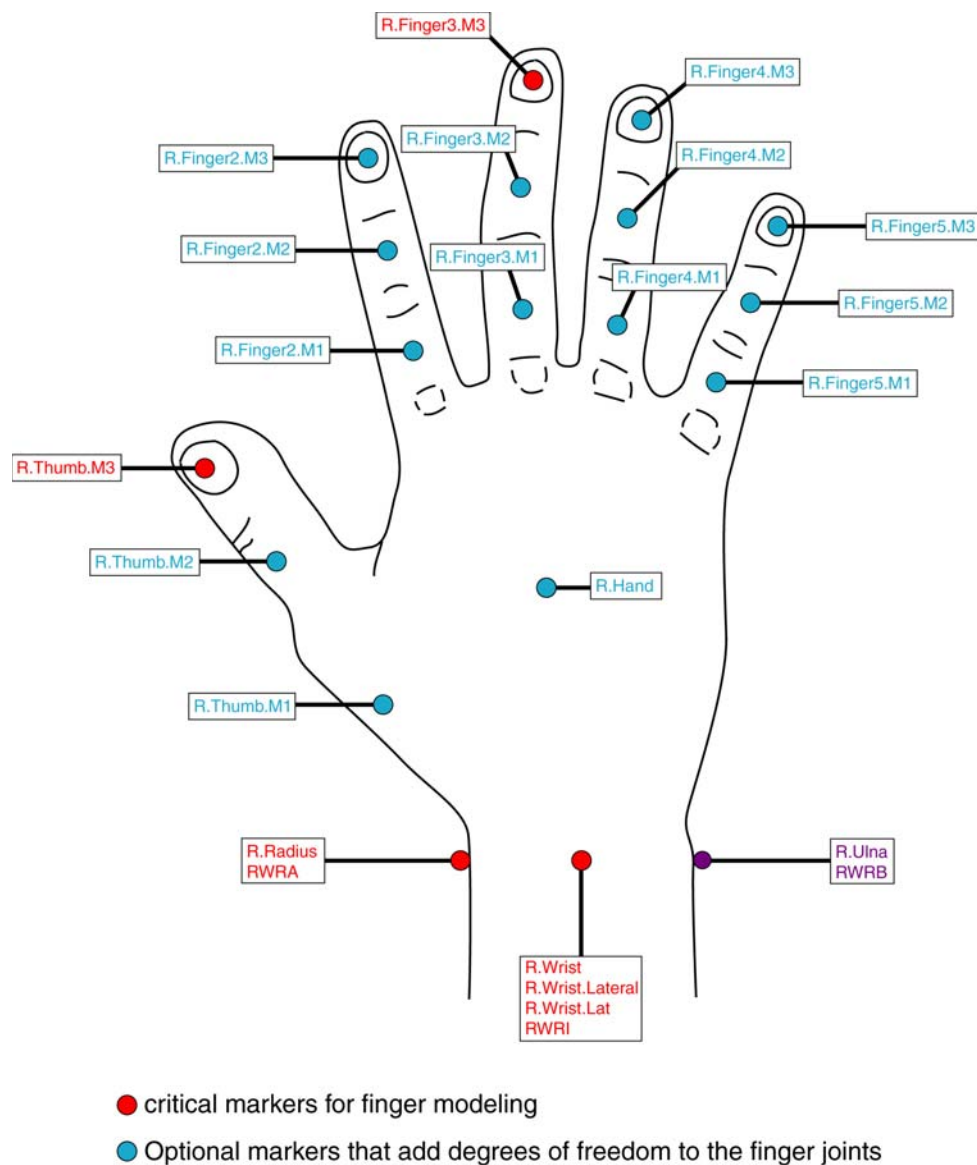
The markers shown in [Figure H-5](#) are optional. If any of these markers is in the static trial, its location on the corresponding body segment in the Mocap Model will automatically be determined after the model has been scaled using the critical markers (*i.e.*, these optional markers are not “fixed,” so their X, Y, Z offsets in the model file will be overwritten when the model is loaded). These markers will then be used to help solve the frames of data in a motion trial.

Figure H-5. Optional Markers



The markers shown in [Figure H-6 on page H-24](#) are used by the Motion Module to control the degrees of freedom in the hand. If the three critical markers are present in the static trial, the Motion Module will load a detailed model of the hand with three joints in each finger. By default, all of the finger joints are fixed. SIMM converts them into hinge joints as it detects the presence of markers to control the joints. For example, if R.Finger2.M1, R.Finger2.M2, and R.Finger2.M3 are all present, SIMM will create three hinge joints in the index finger, each with its own degree of freedom. If only R.Finger2.M1 is present, SIMM will create the proximal finger joint with a degree of freedom, and make the two distal joints dependent on the proximal one (so that all three joints will flex when the proximal one does). Any combination of the optional markers can be used to create a hand model with the desired degrees of freedom. All of the optional hand markers are defined as “fixed” in the model file. This means that the offsets specified in the file are used for solving motions (the Motion Module does not overwrite them), and thus you should place the markers on the subject according to how they are shown in the figure below.

Figure H-6. Critical and Optional Markers for Hands



Starting with SIMM 4.0, support has been added for alternative critical marker sets for use with the Mocap Model. For example, the sacral marker can be replaced with two PSIS markers, and the lateral wrist marker can be replaced with the radius marker. It is thus difficult to display in a single picture of the body the complete set of markers that are required. On the following pages are descriptions of the critical and semi-critical marker sets for each portion of the body. Also, for each marker, the complete list of acceptable names is shown. Any one of these *case-insensitive* names in the list can be used to identify the marker in the **Cortex** project.

Lower Body

The lower body portion of the Mocap Model will be loaded if the critical markers listed below are present in the static trial. The thigh, shank, and feet segments will each be scaled separately, based on measurements made from the static trial. Each of these segments will be scaled uniformly in the X, Y, and Z dimensions. The pelvis segment will be scaled independently in the X, Y, and Z dimensions. It is not possible to load only one leg of the Mocap Model.

Critical Markers

1. **Right ASIS.** acceptable names: R.ASIS RASIS RASI
2. **Left ASIS.** acceptable names: L.ASIS LASIS LASI
3. **Posterior pelvis:**
 - a. **Sacrum.** acceptable names: V.SACRAL V.SACRUM SACRAL SACRUM SACR VSAC
 - or
 - b. **Right PSIS.** acceptable names: R.PSIS RPSIS RPSI
 - and
 - Left PSIS.** acceptable names: L.PSIS LPSIS LPSI
4. **Right lateral knee.** acceptable names: R.KNEE R.KNEE.LAT-
ERAL R.KNEE.LAT RKNE
5. **Left lateral knee.** acceptable names: L.KNEE L.KNEE.LATERAL
L.KNEE.LAT LKNE
6. **Right lateral ankle.** acceptable names: R.ANKLE
R.ANKLE.LATERAL R.ANKLE.LAT RANK
7. **Left lateral ankle.** acceptable names: L.ANKLE L.ANKLE.LAT-
ERAL L.ANKLE.LAT LANK
8. **Right heel.** acceptable names: R.HEEL RHEE
9. **Left heel.** acceptable names: L.HEEL LHEE
10. **Right toe.** acceptable names: R.TOE RTOE
11. **Left toe.** acceptable names: L.TOE LTOE

Semi-critical Markers

1. **Right medial knee.** acceptable names: R.KNEE.MEDIAL
R.KNEE.MED
2. **Left medial knee.** acceptable names: L.KNEE.MEDIAL
L.KNEE.MED
3. **Right medial ankle.** acceptable names: R.ANKLE.MEDIAL
R.ANKLE.MED
4. **Left medial ankle.** acceptable names: L.ANKLE.MEDIAL
L.ANKLE.MED

Upper Body

The upper body portion of the Mocap Model will be loaded if the critical markers listed below are present in the static trial. The upper arm and lower arm segments will each be scaled separately, based on measurements made from the static trial. Each of these segments will be scaled uniformly in the X, Y, and Z dimensions. The torso segment will be scaled independently in two dimensions (the X is scaled the same as the Z). It is not possible to load the upper body with only one arm. To load only one arm (without the rest of the upper body), use the SIMM file rightArm.jnt or leftArm.jnt as the Mocap Model.

Critical Markers

1. **Right ASIS.** acceptable names: R.ASIS RASIS RASI
2. **Left ASIS.** acceptable names: L.ASIS LASIS LASI
3. **Posterior pelvis:**
 - a. **Sacrum.** acceptable names: V.SACRAL V.SACRUM SACRAL SACRUM SACR VSAC

or

Right PSIS. acceptable names: R.PSIS RPSIS RPSI

and

Left PSIS. acceptable names: L.PSIS LPSIS LPSI
4. **Right shoulder.** acceptable names: R.SHOULDER RSHO
5. **Left shoulder.** acceptable names: L.SHOULDER LSHO
6. **Right lateral elbow.** acceptable names: R.ELBOW R.ELBOW.LATERAL R.ELBOW.LAT RELB
7. **Left lateral elbow.** acceptable names: L.ELBOW L.ELBOW.LATERAL L.ELBOW.LAT LELB
8. **Right wrist:**
 - a. **Lateral.** acceptable names: R.WRIST R.WRIST.LATERAL R.WRIST.LAT RWRI

or

b. Radius. acceptable names: R.RADIUS RWRA
9. **Left wrist:**
 - a. **Lateral.** acceptable names: L.WRIST L.WRIST.LATERAL L.WRIST.LAT LWRI

or

b. radius. acceptable names: L.RADIUS LWRA

Semi-critical Markers

1. **Right medial elbow.** acceptable names: R.ELBOW.MEDIAL R.ELBOW.MED
2. **Left medial elbow.** acceptable names: L.ELBOW.MEDIAL L.ELBOW.MED
3. **Right wrist:**
 - a. **Medial.** acceptable names: R.WRIST.MEDIAL R.WRIST.MED

or

b. ulna. acceptable names: R.ULNA RWRB

4. Left wrist:

a. medial. acceptable names: L.WRIST.MEDIAL L.WRIST.MED

or

b. ulna. acceptable names: L.ULNA LWRB

Right Arm

To load only the right arm, set the MOCAP_MODEL parameter in your SIMM preferences file to *rightArm.jnt*, or choose that file using the **Options...Choose Model Model** command in the SIMM menu bar. Then use the markers listed below.

Critical Markers

1. **Right shoulder.** acceptable names: R.SHOULDER RSHO
2. **Right lateral elbow.** acceptable names: R.ELBOW
R.ELBOW.LATERAL R.ELBOW.LAT RELB
3. **Right wrist:**
 - a. Lateral.** acceptable names: R.WRIST R.WRIST.LATERAL
R.WRIST.LAT RWRI

or

b. Radius. acceptable names: R.RADIUS RWRA

Semi-critical Markers

1. **Right medial elbow.** acceptable names: R.ELBOW.MEDIAL
R.ELBOW.MED
2. **Right wrist:**
 - a. Medial.** acceptable names: R.WRIST.MEDIAL R.WRIST.MED

or

b. Ulna. acceptable names: R.ULNA RWRB

Left Arm

To load only the left arm, set the MOCAP_MODEL parameter in your SIMM preferences file to *leftArm.jnt*, or choose that file using the **Options...Choose Model Model** command in the SIMM menu bar. Then use the markers listed below.

Critical Markers

1. **Left shoulder.** acceptable names: L.SHOULDER LSHO
2. **Left lateral elbow.** acceptable names: L.ELBOW L.ELBOW.LATERAL
L.ELBOW.LAT LELB
3. **Left wrist:**
 - a. Lateral.** acceptable names: L.WRIST L.WRIST.LATERAL
L.WRIST.LAT LWRI

or

a. Radius. acceptable names: L.RADIUS LWRA

Semi-critical Markers

1. **Left medial elbow.** acceptable names: L.ELBOW.MEDIAL L.ELBOW.MED
2. **Left wrist:**
 - a. **Medial.** acceptable names: L.WRIST.MEDIAL L.WRIST.MED

or

 - b. **Ulna.** acceptable names: L.ULNA LWRB

Right Hand

The right hand will always be included when the right arm is loaded, even if there are no markers on the hand. The presence of critical markers controls how the hand is scaled and what degrees of freedom it has. The right hand will be scaled separately from the right lower arm if the three critical markers listed below are present in the static trial. The individual finger gencoords will be added to the model if the three critical hand markers and the appropriate finger markers are present in the static trial.

Critical Markers

1. **Right thumb.** acceptable names: R.THUMB R.THUMB.M3
2. **Right middle finger.** acceptable names: R.MIDDLE.FINGER R.FINGER R.FINGER3.M3
3. **Right wrist:**
 - a. **Lateral.** acceptable names: R.WRIST R.WRIST.LATERAL R.WRIST.LAT RWRI

or

 - b. **Radius.** acceptable names: R.RADIUS RWRA

Semi-critical Markers

1. **Right wrist:**
 - a. **medial.** acceptable names: R.WRIST.MEDIAL R.WRIST.MED

or

 - b. **Ulna.** acceptable names: R.ULNA RWRB

Left Hand

The left hand will always be included when the left arm is loaded, even if there are no markers on the hand. The presence of critical markers controls how the hand is scaled and what degrees of freedom it has. The left hand will be scaled separately from the left lower arm if the three critical markers listed below are present in the static trial. The individual finger gencoords will be added to the model if the three critical hand markers and the appropriate finger markers are present in the static trial.

Critical Markers

1. **Left thumb.** acceptable names: L.THUMB L.THUMB.M3
2. **Left middle finger.** acceptable names: L.MIDDLE.FINGER L.FINGER L.FINGER3.M3
3. **Left wrist:**
 - a. **Lateral.** acceptable names: L.WRIST L.WRIST.LATERAL L.WRIST.LAT LWRI

or

b. Radius. acceptable names: L.RADIUS LWRA

Semi-critical Markers

1. Left wrist:

a. Medial. acceptable names: L.WRIST.MEDIAL L.WRIST.MED

or

b. Ulna. acceptable names: L.ULNA LWRBb

Head

The head will always be included when the upper body is loaded, and the neck will contain three degrees of freedom. If the critical markers listed below are present in the static trial, the head will be scaled separately from the torso. Otherwise, the head will be scaled uniformly by the scale factor used for the Y (height) of the torso. If no markers (critical or optional) are included on the head in the static trial, then the degrees of freedom in the neck will remain fixed during imported motions.

Critical Markers

1. Rear of head. acceptable names: HEAD.REAR REAR.HEAD HEADREAR REARHEAD

2. Top of head. acceptable names: HEAD.TOP TOP.HEAD HEAD-TOP TOPHEAD

3. Front of head. acceptable names: HEAD.FRONT FRONT.HEAD HEADFRONT FRONTHEAD

The following optional markers are already defined in **mocap.jnt**. To use them, just put their exact names in your **Cortex** project:

pelvis: R.Trochanter
L.Trochanter

thorax: Offset
Sternum
T10
CLAV
STRN
RBAK

cerv7: C7
C7 Spinous Process

head: R.Ear
L.Ear
RBHD
RFHD
LBHD
LFHD
HEDO
HEDP
HEDA
HEDL

clavicle_l: L.Clavicle

scapula_l: L.Scapula

L.Scapula.Top

L.Scapula.Bottom

L.Angulus Acromialis

L.Trigonum Spinae

L.Angulus Inferior

humerus_l: L.Bicep

L.Biceps.Lateral

ulna_l: L.Forearm

femur_l: L.Thigh

L.Thigh.Upper

L.Thigh.Front

L.Thigh.Rear

LTHI

tibia_l: L.Shank

L.Shank.Upper

L.Shank.Front

L.Shank.Rear

LTIB

foot_l: L.MedFoot

L.LatFoot

clavicle_r: R.Clavicle

scapula_r: R.Scapula

R.Scapula.Top

R.Scapula.Bottom

R.Angulus Acromialis

R.Trigonum Spinae

R.Angulus Inferior

humerus_r: R.Bicep

R.Biceps.Lateral

ulna_r: L.Forearm

femur_r: R.Thigh

R.Thigh.Upper

R.Thigh.Front

R.Thigh.Rear

RTHI

tibia_r: R.Shank

R.Shank.Upper

R.Shank.Front

R.Shank.Rear

RTIB

foot_r: R.MedFoot

R.LatFoot

Synchronizing Digital Video with Cortex

Topic	Page
EVaDV Overview	I-1
System Requirements	I-1
Installation	I-2
Using EVaDV	I-2
Currently Open Known Issues	I-3

EVaDV Overview

EVaDV is a Digital Video (DV) capture application for use with Motion Analysis Corporation's **Cortex** software for the synchronized capture of color video data on a separate Windows PC as *.avi files. You can directly transfer digital information back and forth between a DV camcorder and your computer with the use of the IEEE 1394 standard, also known as a Firewire or i.Link connection. If your computer does not come with this interface built into it, you will need to purchase an inexpensive card that provides the correct port.

You can run **EVaDV** on your local machine that has **Cortex** running on it (not recommended) as well as with remote machines that are connected to digital video cameras.

Note: Any standard DV camcorder should be sufficient for use with the **EVaDV** software. All **EVaDV** testing and product development was done with a Sony™ DCR-TRV520 NTSC model DV camcorder.

System Requirements

Recommended Minimum Specifications

- **Microsoft Windows XP or Vista**
- 256 MB RAM
- 80GB (or more) free hard drive space (for captured files)
- CD-RW drive
- Ethernet card

Installation

We recommend you install the software into the **C:\Program Files\Motion Analysis\EVaDV** directory, but **EVaDV.exe** will run from any folder.

Note: No dongle or license file is required to run this application. But to collect synchronized color video (AVI files) in **Cortex**, you need the [Reference Video 3.0] line in your mac_lic.dat license file.

Using EVaDV

Capturing Digital Video Using EVaDV

1. Launch **EVaDV**.
2. Select the desired camera/capture device from Camera dropdown. If there is a single video camera connected to the host system, **EVaDV** connects to this camera automatically.
3. Select a Capture Folder to indicate where captured files should be stored.
4. Select a Capture File to indicate the name of the file that will be created.
5. To begin recording, press the **Record** button (red circle). A red RECORD will display next to the player control panel indicating the system is recording, and a message will be displayed in the Message Bar indicating record start time.
6. To stop recording, press the **Stop** button. The red RECORD will disappear and a message will be displayed in the Message Bar indicating recorder filename, stop time, and any available stats.

Capturing Digital Video in EVaDV from Cortex

Note: [EvaXX] indicates the host machine on which to perform the action.

1. Ensure the **EVaDV** host and the **Cortex** host are connected to each other via a TCP/IP network connection.
2. [EvaDV] Launch **EVaDV** on the video capture host.
3. [Cortex] Launch **Cortex** on the **Cortex** host.
Note—launch order is unimportant, however if **Cortex** grabs the camera/capture device first (i.e. displays color video window) **EVaDV** will not have access to the camera. **Cortex** will grab the camera when you press the **F1** key in a capture window and release it when you select another key (**F2**, **F3**, etc.).
4. [EvaDV] Select the desired camera/capture device from Camera dropdown. If there is a single video camera connected to the host system, **EVaDV** connects to this camera automatically.
5. [Cortex] Activate the Color Video (*.avi) checkbox in the **Motion Capture > Output** panel. Press the **F1** key to show video and grab the DV camera.
6. [Cortex] Press the **Record** button.

7. **[EVaDV]** On Record, **EVaDV** will record the file specified by **Cortex** to the directory specified by **Cortex** or, if not present, to the local Capture Folder. The Message Bar indicates the directory requested by **Cortex**.
8. **[EVaDV]** On Record, A red "RECORD" will display next to the player control panel indicating system is recording, and a message will be displayed in the Message Bar indicating record start time.
9. **[Cortex]** Stop recording by pressing either the **Stop** button or after reaching the duration specified.

Note: If you have more than one computer running **Cortex** on the same network, the **EVaDV** software does not consume the plugin port (as does the streaming Alias/Motionbuilder online plugin, for example). The message about starting and ending the data capture is broadcast to the x.x.x.255 address which means that all computers connected can hear the message and start and stop the recording. So you should be able to connect as many **EVaDV** recorder-computers up as you need.

Note: You probably will not want the AVI files streaming across the network.

Note: When VC files are collected, AVI files are collected. If VC files are not collected, AVI files are not collected.

Note: You must have the color video window (F1 function) open in the **Cortex** interface to record AVI files.

Suggestion for DV Camera Setup

Sony™ DV cameras have a setup mode called **FRAME/FIELD**. The **FRAME** setup mode works best as it eliminates the "motion blur" which results from the **FIELD** (also known as Interleaved) mode. On Canon™ DV cameras, you will want to select **MOVIE** mode.

Currently Open Known Issues

- Interface is evolving.
- Resizing video window by dragging corner can cause a hang-up. Stop, resize, play is a resolution for this.
- File recompression functionality unstable with certain compressors.
- Multiple camera support is working, however certain cameras (Sony) dislike it, and will give a "fail on run" message.
- Snapshot **.tga** image is incorrect.
- Only Type-1 (interleaved) AVI files are supported (import/export).

Known Graphic Card Issues

There is a known problem displaying the AVI files when a 3D window is displayed on a Windows XP system. The AVI file plays in a jerky motion and the screen is sometimes sliced into horizontal blocks.

This is a problem with Open GL and Direct X displaying at the same time. Right-click on your desktop, then select **Properties > Settings > Advanced > Troubleshoot**. Slide the Hardware Acceleration down a few points. Quit **Cortex**, then re-launch. See if that fixes the problem. The settings depend on what your graphics card does and has for features that

changes from one computer to another. If that does not fix the problem, try to slide the Hardware Acceleration to **None**. Quit **Cortex** and then re-launch. If that still does not fix the problem, try to change some of the Open GL settings found under the name of your graphics card (found under the **Properties > Setting > Advanced** tab). Also, change the Vertical Sync setting to **On by Default** and restart your computer.

Using Cortex with Jack Software

Topic	Page
Introduction	J-1
Installation Instructions	J-2
Jack – Using the Motion Analysis Calcium Tracker Module	J-2
Jack5.0a Required & Recommended Marker Sets in Cortex for Calcium Skeleton Generation	J-5

Introduction

Jack is an ergonomics and human factors product used in various industries to improve the ergonomics of product designs and workplace tasks. This software enables users to position biomechanically-accurate digital humans of various sizes in virtual environments, assign them tasks and analyze their performance. Jack (and Jill) digital humans can tell engineers what they can see and reach, how comfortable they are, when and why they're getting hurt, when they're getting tired and other important ergonomics information. This information helps organizations design safer and more effective products faster and for less cost. Ultimately, Jack helps companies bring factories on-line faster and optimize productivity while improving worker safety. For more information, please visit the UGS website at www.ugs.com.

Before You Start

Jack 5.0a does not need a new license file. It uses the same license as Jack 4.2. The license is based on your Ethernet card's Physical Address. To find it:

1. Select **Start > Run**. This command. brings up a DOS command window.
2. Type `ipconfig /all`
3. Find the line that looks like:
Physical Address 000-50-DA-8E-BC-BD
<<--
4. Note yours and then contact UGS.

Installation of the Jack5.0a software requires very specific installation as per the instructions in the next section. If you deviate from the settings specified it will not launch.

The .jk.tcl file will not easily copy itself to a native windows environment. This will be fixed for the release version of Jack 5.0. The .jk.tcl acts as a launching tool for the Calcium plugin window. Currently, copying this file from a web-based browser (like Hotmail or Gmail) will allow you to save it as a .jk.tcl file (MS Windows thinks it doesn't have a filename). If you are using an MS product or Novell Webmail, it will try to rename it to .jk.tcl[1] and you cannot rename it. If you open the .jk.zip folder and extract the file, it will copy itself correctly, without appending the [1] to the end of the filename.

Installation Instructions

1. Download the latest Jack-5.0alpha from ftp://specialdevftp.eai.com/private/jack_dist/FordDW/Jim/Jack50a
2. Unzip each zip file to the **C:\Jack50a** folder.
3. Copy your Jack license file to the **C:\Jack50a\license** folder.
4. Copy the attached jk.tcl file to the **C:\Documents and Settings\<your_login_id>** folder.

Note: If you have a HOME environment variable set on your machine, copy the .jk.tcl file to your HOME folder.

5. To start Jack, double click on the **jack50.bat** file in the **C:\Jack50a** folder.
 - For debugging, you can use the **Start > Run...** command in a DOS window. Then `cd \Jack50a` and run the batch file: **jack50.bat**. This gives more information.

Jack – Using the Motion Analysis Calcium Tracker Module

Loading CalciumTracker module in Jack

1. Start Jack.
2. Click **Modules | Plug-ins...**
3. In the Add-On Modules Dialog, select the **CalciumTracker** module.
4. Click **Load** and **OK**.
5. Click **Modules | CalciumTracker | MotionAnalysis**.

-OR-

1. Start Jack (auto loading of Calcium Tracker occurs via the .jk.tcl script).
2. *[Optional]* If you are using .jk.tcl file, click **Modules | CalciumTracker Dialog**.

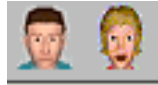
Connecting to MotionAnalysis

1. In the **MotionAnalysis-Calcium** dialog, click on the **Devices** tab. Enter the **Cortex** or localhost computer name as the Host name, and then click **Connect**.
2. Check/Uncheck **Display Bodies** to turn on/off the visible bodies in Jack.
3. Set the Origin Rotation Check box to **ON**, select **X Rotation –90**, **Z Rotation –90**.

Auto-Scaling, Constraining and Positioning the Subject

1. Create a male or female Jack figure.
This may be done by selecting either of the following buttons.

Figure J-1. Male or Female Selection Button



2. Click the **pick** button and select the subject from the scene.
The subject turns yellow when it is picked. Left-click on the subject.
The name "Human" should fill the blank box beside the pick icon.
3. Click on the **Subject** tab.
4. Have the actual subject in standing straight posture and click **Auto Scale**.
5. Click **Constrain**.
6. Click **Move**. Using the mouse in the Jack window, move the MA_ORIGIN figure to position the subject.

Creating Two Channel Eye View Windows (First Person) [OPTIONAL]

1. In the MotionAnalysis-Calcium dialog, click the **pick** button and select the subject from the scene.
2. Click on the **Eye View** tab. Click **Apply** (minimize all other windows). Note that the main Jack window is disabled. To enable, right - middle button on the TJ_Window and uncheck **Disabled**.
3. *[Optional]* You can change eye view window parameters.

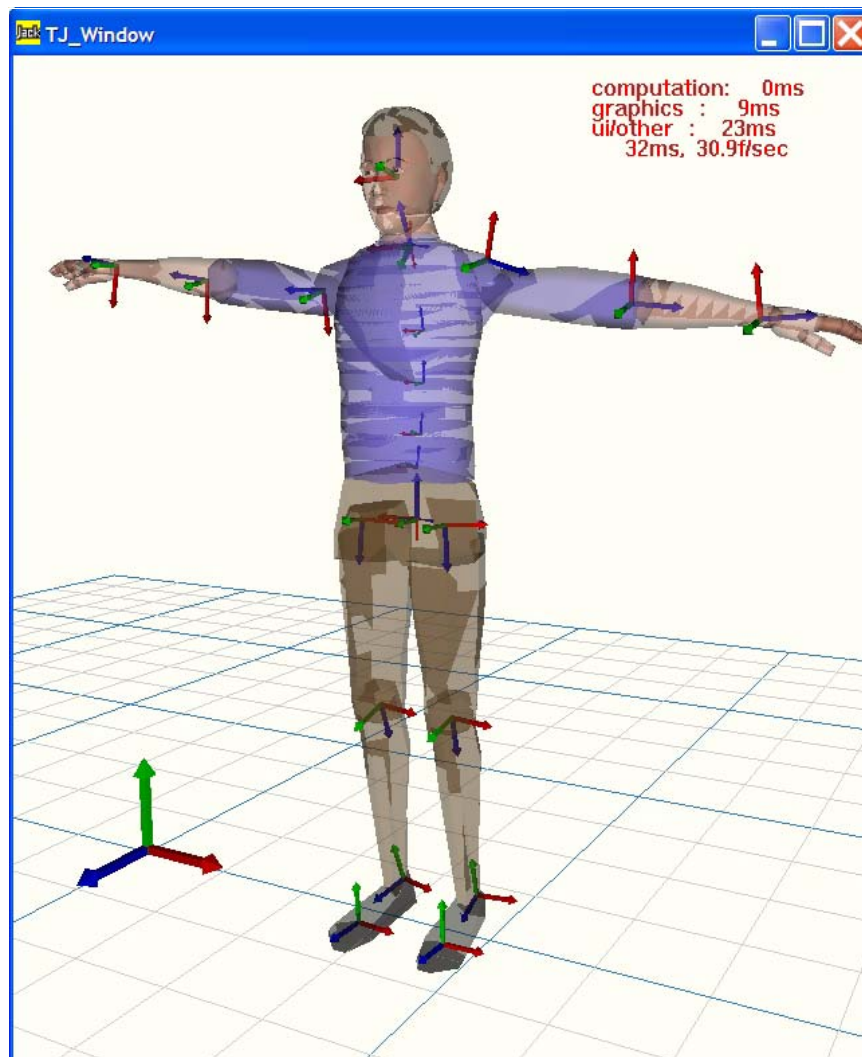
Setting Up Collision Detection [OPTIONAL]

1. In the MotionAnalysis-Calcium dialog, click the **pick** button and select the subject from the scene.
2. Click on the **Collision** tab. Check the required segments and select an algorithm for the collision check. Then click **Apply**.

Note: You should have a scene loaded before you setup the collision detection.

**List of Segments
Required by
Jack**

1. Root
2. Head
3. Neck
4. Spine1/Spine2/Spine3/Spine4
5. LClavicle/RClavicle
6. LUpperArm/RUpperArm
7. LLowerArm/RLowerArm
8. LHand/RHand
9. LHip/RHip
10. LUpperLeg/RUpperLeg
11. LLowerLeg/RLowerLeg
12. LFoot/RFoot
13. LToes/RToes

Figure J-2. Jack Interface

Jack5.0a Required & Recommended Marker Sets in Cortex for Calcium Skeleton Generation

Table J-1. Required markers for Jack 5 SIMM-OrthoTrak Model: 22 (absolute minimum)

Markers	Location
R.Shoulder, L.Shoulder	markers on top of shoulder above shoulder joint
R.Elbow, L.Elbow	lateral side of elbow, on elbow hinge axis. On top of elbow with arms in T-pose position
Wrist has 2 options: R.Radius, L.Radius RECOMMENDED (on distal radius, thumb side of your hand), and R.Ulna, L.Ulna (on distal ulna, pinky side of your hand) -and/or- R.Wrist, L.Wrist: top of wrist	
R.ASIS and L.ASIS	Anterior Superior Iliac Spine
L.BackOffset	offset marker for asymmetry
Back of pelvis has 2 options: R.PSIS and L.PSIS: RECOMMENDED Posterior Superior Iliac Spine -and/or- V.Sacral (All 5 is RECOMMENDED: ASIS, PSIS, V.Sacral)	
R.Knee and L.Knee	lateral knee, close to knee axis
R.Ankle and L.Ankle	lateral ankle, on fibular malleolus
R.Heel and L.Heel	heel, at same height as toe markers
R.Toe and L.Toe	toe markers, center of foot at proximal base of toe joint

The following markers are a recommended marker set only. These incorporate the above markers, include some additional markers, and match the data set Jack5_41Markers_AutoScale.prj in the Sample Data files.

1. Top.head
2. Back.Head
3. Front.Head
4. L.Head_Offset
5. R.Shoulder
6. L.Shoulder
7. Neck
8. L.BackOffset
9. R.Bicep
10. R.Elbow
11. R.ForeArm
12. R.Radius
13. R.Ulna
14. R.Thumb
15. R.Pinky
16. L.Bicep
17. L.Elbow
18. L.Forearm
19. L.Radius
20. L.Ulna
21. L.Thumb
22. L.Pinky
23. R.ASIS
24. L.ASIS
25. R.PSIS
26. L.PSIS
27. V.Sacral
28. R.Thigh
29. R.Knee
30. R.Shank
31. R.Ankle
32. R.Heel
33. R.Toe
34. R.Foot
35. L.Thigh
36. L.Knee
37. L.Shank
38. L.Ankle
39. L.Toe
40. L.Heel
41. L.Foot

Jack 5 Marker Set

Figure J-3. Front View

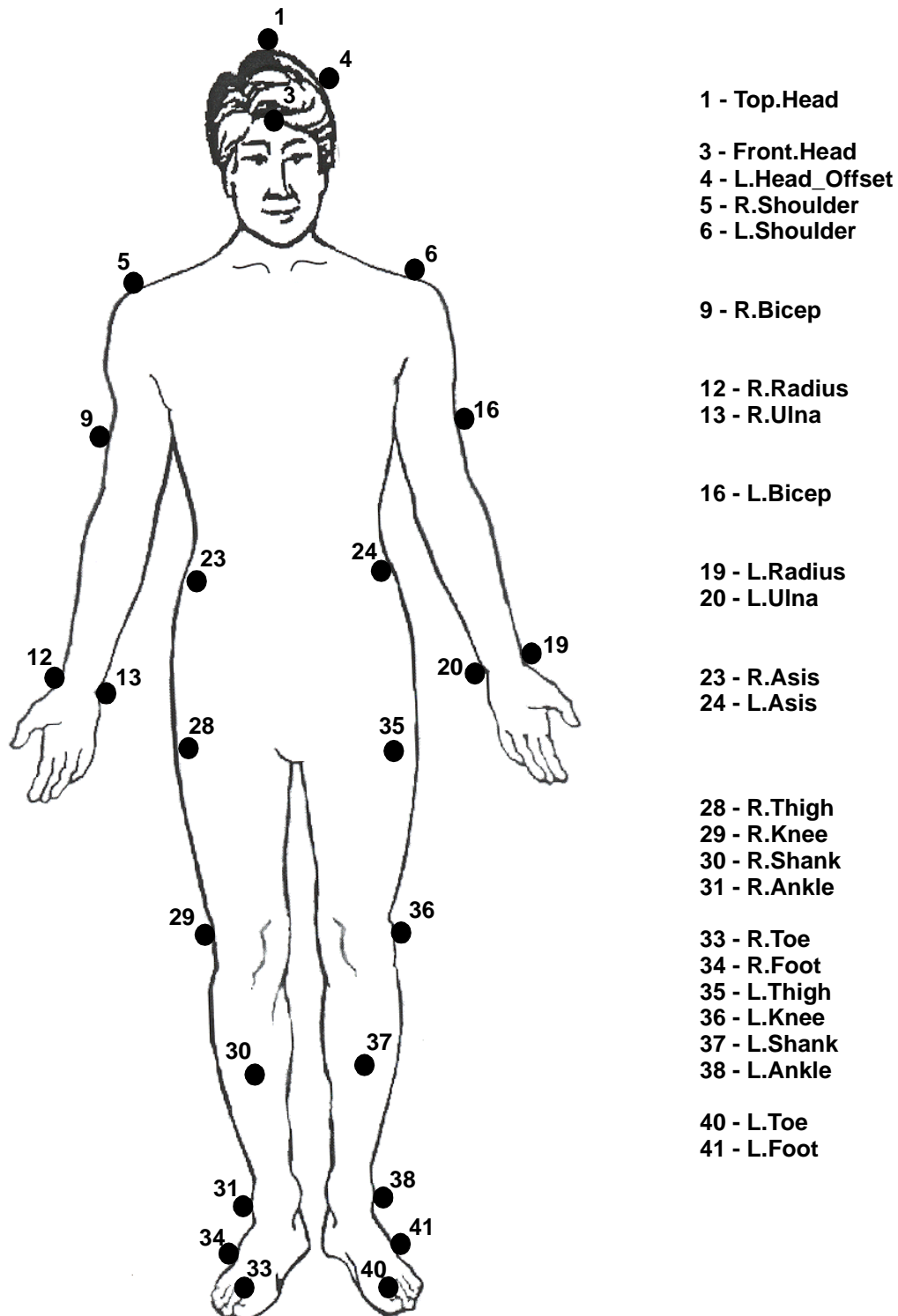


Figure J-4. Right Side View

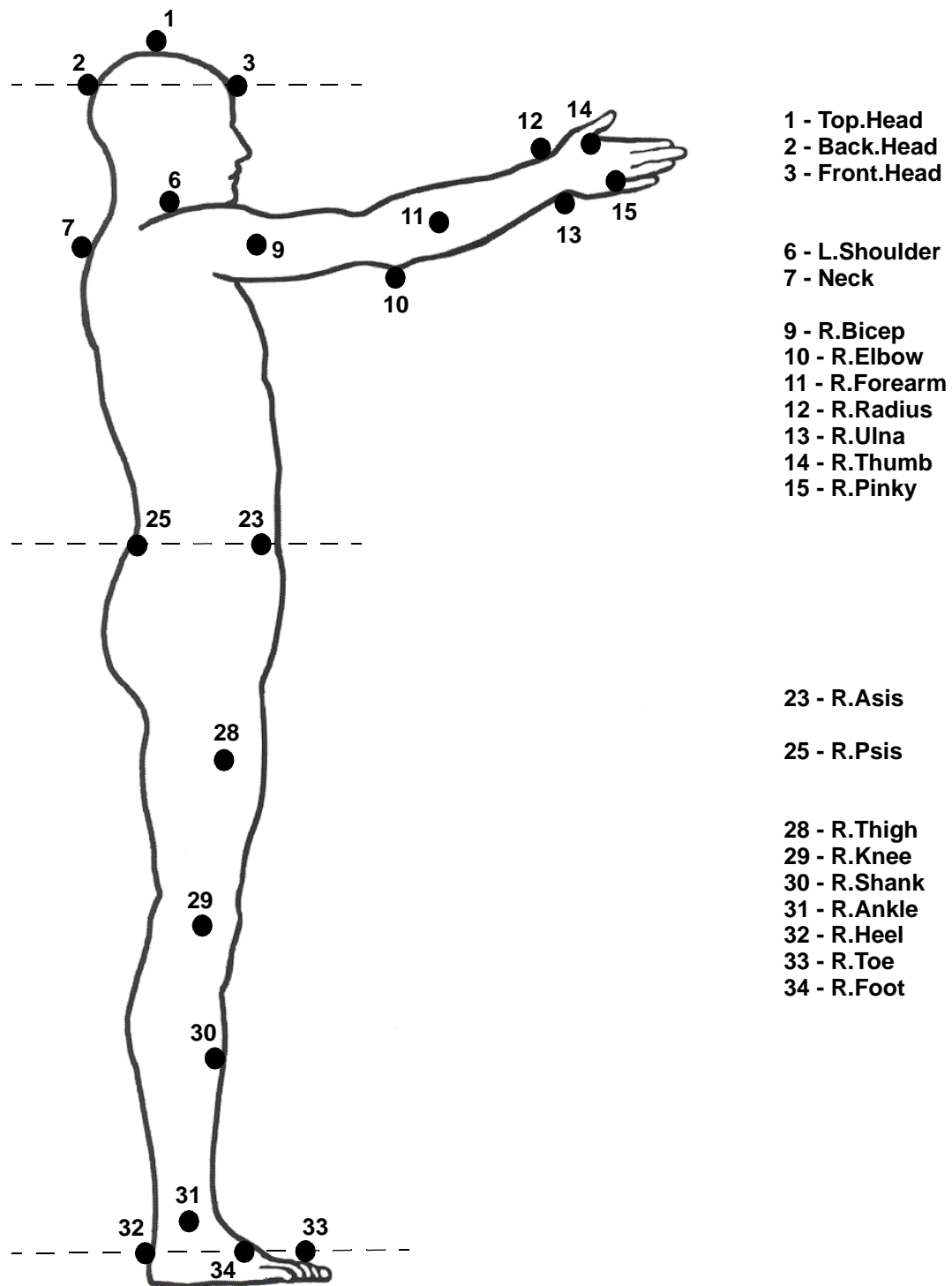


Figure J-5. Left Side View

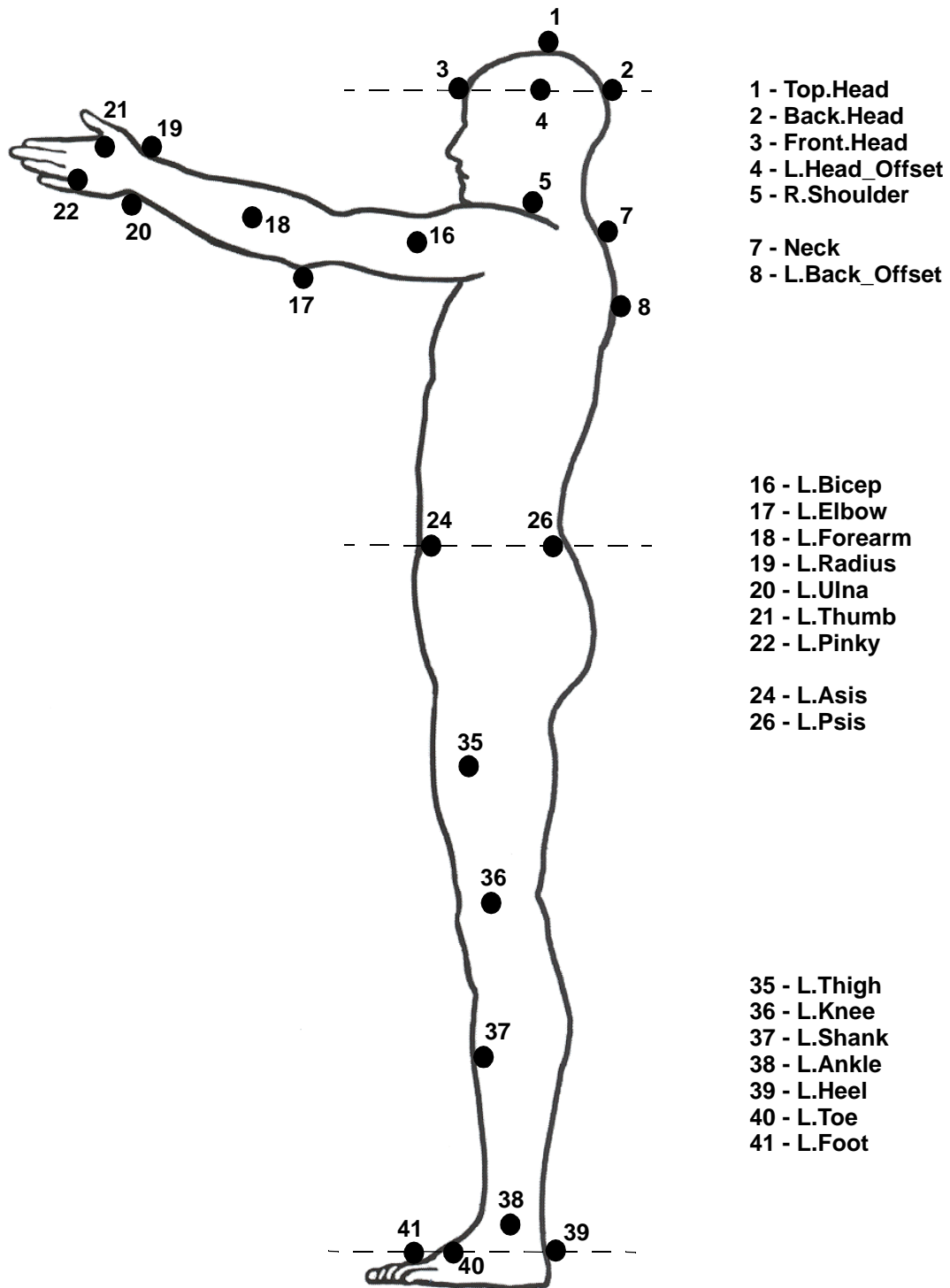
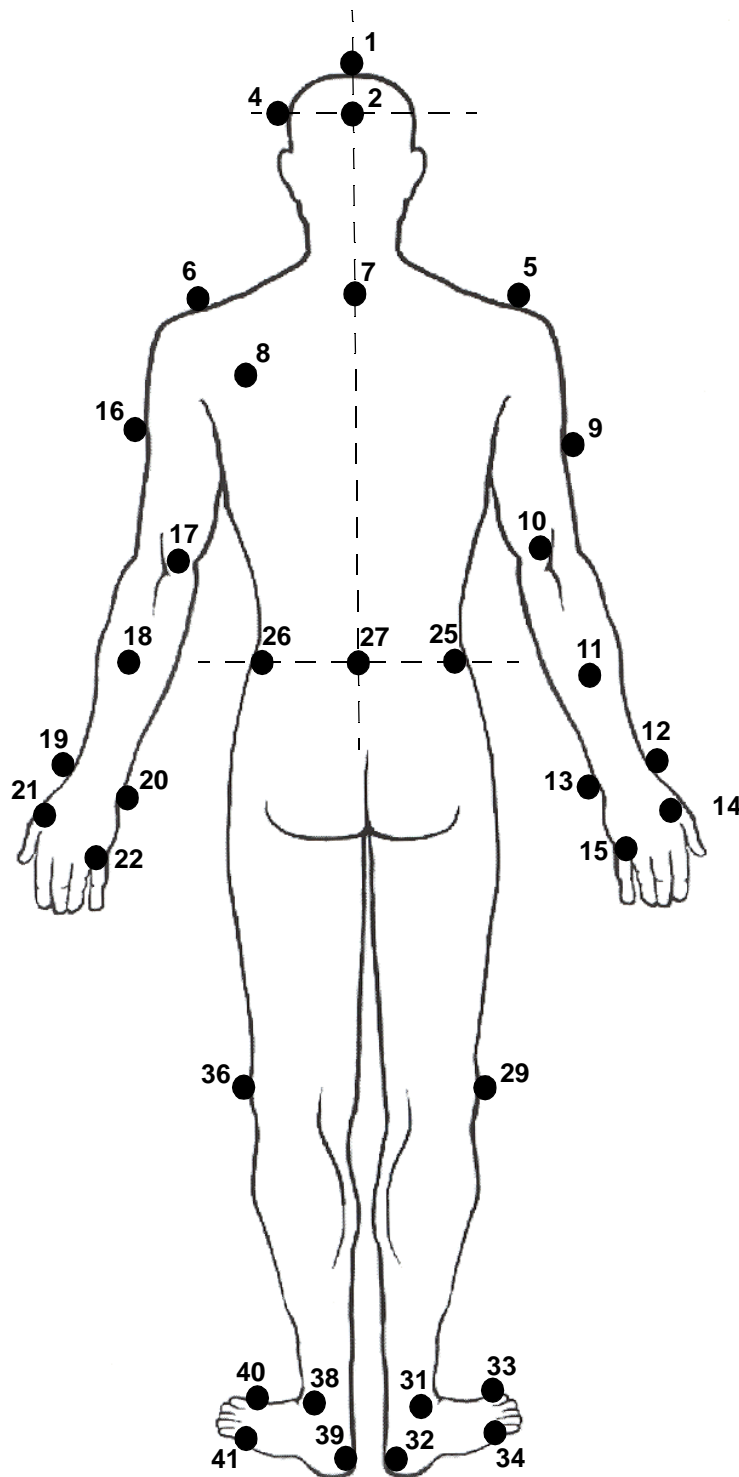


Figure 15. Rear View



- 1 - Top.Head
- 2 - Back.Head
- 4 - L.Head_Offset
- 5 - R.Shoulder
- 6 - L.Shoulder
- 7 - Neck
- 8 - L.Back_Offset
- 9 - R.Bicep
- 10 - R.Elbow
- 11 - R.Forearm
- 12 - R.Radius
- 13 - R.Ulna
- 14 - R.Thumb
- 15 - R.Pinky
- 16 - L.Bicep
- 17 - L.Elbow
- 18 - L.Forearm
- 19 - L.Radius
- 20 - L.Ulna
- 21 - L.Thumb
- 22 - L.Pinky
- 25 - R.Psis
- 26 - L.Psis
- 27 - V.Sacral
- 29 - R.Knee
- 31 - R.Ankle
- 32 - R.Heel
- 33 - R.Toe
- 34 - R.Foot
- 36 - L.Knee
- 38 - L.Ankle
- 39 - L.Heel
- 40 - L.Toe
- 41 - L.Foot

Appendix K *Questions and Answers for Specific Applications*

Question *Does **Cortex** require a specific order for markers and linkages?*

Answer For the identification of markers to work swiftly in real-time the order of marker definitions is important. You should follow these rules.

1. Markers should be ordered such that each successive marker builds the character top to bottom through linkages (i.e. Head to Neck, down one arm, then down the other, down the torso to the hips, down one leg, then the other). Do not backtrack.
2. If the first markers are linked into a stiff triangle, marker identification will be swift. For this reason, the head markers should always be first. Linkage order may affect the rectify process. For the head, a linkage order of 1-2, 1-3, and 2-3 works well.

Question *How can I prevent ghost markers from appearing?*

Answer

1. Ghost markers may appear if the Max. Residual value is set too low. This parameter is set in the tracking function in the Motion Capture mode (except for centroid function instead of tracking).
2. Set the minimum number of cameras to 3.
3. Increase the minimum number of lines per marker

Question *How do I control the length of a recorded file in **Cortex**?*

Answer All the recording options are set in the Output function in the Motion Capture mode.

When the record button is clicked using the trigger or the mouse, a new file is recorded and saved with the name, directory, and output type(s) you have specified. The recording will stop when either:

- the duration in seconds is reached
- the **Stop** button is clicked using the trigger or mouse

The default duration is 60 seconds. If you always want to control the end of the take with the trigger or mouse, we recommend setting the duration to a number that is higher than the trials you usually capture such as 10 minutes (600 seconds).

Question *Can **Cortex** be used in a large capture volume (for example 50'x50', 50 cameras)?*

Answer If care is taken during setup, motion capture will work well in a large volume. The four areas requiring attention are:

1. **Camera Setup**—Are your cameras covering your volume efficiently? Are you using an overlapping volume setup? If not, you may have too many cameras seeing the same area. Over coverage can result in an over abundance of data, slowing the system down.

Note: Try to have no more than three cameras see any given area of the capture volume from one direction and 10 to 12 cameras total.

2. **Calibration**—Tracking residuals should be below 2.0 mm. If not, try raising the Max. Residual value. Too low a value may cause ghost markers to appear.
3. **Template**—Verify that marker identity is being performed quickly. Click the **Reset IDs** button several times while the actor is in the capture space and see if there is a lag in acquiring marker identification. If there is a lag, you may need to create a better template. Verify that the first three markers are the head markers and that the first three links form a rigid triangle. Finally, verify that remaining link definitions flow down the body following the marker definitions.
4. **Frame to Frame Rectifying**—This is mainly influenced by your 3D data quality and tracking parameters. Too many extra, stretchy link-ages can cause problems here.

Question *Can I use MoCap Solver, Si 2.0, or Calcium in **Cortex**?*

Answer Yes. Export a MOD file from Si/Calcium and name it the same as the project file. Select **Model Edit > Tree View** and then select **Calcium Solver** in the Skeleton Engine field.

Question *What is the order of the data in the TRC or TRB files when you use MTOs for tracking?*

Answer The resulting **.trc** file from MTO tracking should match the marker order of the resulting "Merge MarkerSets" operation.

Question *If I have a Solver skeleton setup from **EVa 6.x**, can I use it in **Cortex**?*

Answer Yes, if you go to the **Model Edit > Tree View** panel and in the Skeleton Engine field, you check **Calcium Solver**. in **Cortex**. You must also copy the MOD file into your current project folder with the same name as your current PRJ file.

Question	<i>If I have the EVa 6.X Solver setup, can I get an HTR file from Cortex?</i>
Answer	Yes. Select the Calcium Solver Skeleton Engine button (Bone button, in lower left of the Post Process dashboard), then select File > Export HTR File....
Question	<i>What are the tradeoffs in capturing at 120 fps or faster or slower with Eagle cameras?</i>
Answer	With the older analog cameras, there were tradeoffs in image quality as the frame rate went up. With the Eagle cameras, we see no degradation AT ALL with the higher frame rates, which is great for high speed captures. The images are all taken off of the Eagle sensors in 2 msec which corresponds to the 500 fps. Waiting longer between frames does not degrade or enhance the image quality. The only considerations are Ethernet bandwidth (not generally a problem), disk space used, and time to process (or post-process) the raw VC files.
Question	<i>Are there any rules that should be followed when deciding which camera should be set as "Master"?</i>
Answer	Any camera can be master. If you have an A-D system, the master camera must be connected to it using the A-D sync cable.
Question	<i>When you see a marker in the 2D display, are you simply seeing a digital representation of what the camera sees at the CCD, or are any of the tracking parameters incorporated into determining whether the system "sees" a marker (i.e. marker size, centroid parameters, etc.)?</i>
Answer	The black data is the raw edge data, affected only by the lighting and the Threshold. The red dots (lens corrected and/or not) are the calculated centroids. To calculate a centroid, there are two main things: 1) Min Lines per Marker (usually set to 2 or 3 lines), 2) Max. lines per Marker (usually set to a BIG number like 100), and Shape Analysis (None, Normal or Weak), normally to Normal. But sometimes it is set it to None if it is tossing out centroids, like during a L-Frame seed calibration.
Question	<i>Can you connect 7 or 8 cameras to an EagleHub? We tried connecting 8 cameras to an EagleHub2 and the data transfer to the gigaswitch became quite unstable. Note that this pertains to the older 8-port EagleHubs only (not the 12-port EagleHubs).</i>
Answer	There are only 8 useful ports on an older model EagleHub, which means 7 cameras can be connected to the EagleHub, and one more is used for the uplink to the Network Interface Card (NIC). An eighth camera can be connected directly to one of the remaining open port son the NIC using a patch cable.

Question *Is there a way to have two templates for two people in the project that identify them both in real time at once, rather than using one huge template that includes everything?*

Answer This is the MTO (Multiple Tracking Object) item. Refer to [“Multiple Tracking Objects” on page 9-7](#).

Question *What does the extend template option do?*

Answer Extend Template adds new linkage stretch to the existing template if you need it. For example, you can make a one frame template, ID some motion, then you can extend the template so that it knows about the new motion linkage stretches as well as the old ones.

Question *What are the latest specifications for the Eagle system performance?*

Answer From the 5 person Eagle camera data included in the **Cortex** release package under the Samples folder:

- Trial: FivestarsAgainandAgain.VCX: 1800 Frames, 120 Frames/sec. per the Cortex software.
- Biggest VC file: 3900 KB (highest data rate), Smallest VC File: 1100 KB, Avg. VC. size about 2500 KBytes.
- Data rate per Eagle camera: Max size file: 3900 KBytes X 120 Frames/sec X 1/1800 Frames = 260 KBytes/sec or about 2600 Kbps (kilobits/sec) or 2.6 Mbps or about 2.6% of the 100 Mbps Ethernet or about 0.26% of the 1000 Mbps Ethernet
- Avg. size file: 2500 KBytes X 120 Frames/sec X 1/1800 Frames = 166 KBytes/sec or about 1660 Kbps or about 1.6% of the 100 Mbps Ethernet

So, what does this mean for your 12 Camera Setup?

For a 5 person Eagle camera capture, with an average data rate for 120 Hz capture per camera, it works fine (about 1660 Kbps X 12 cameras = 19,920 Kbps or about 20% of the available Ethernet bandwidth). We used a Gigabit Ethernet NIC and Switch for our 24 camera setup. That used about 4% of the 1000 Mbps Ethernet, but would have been 40% of the 100 Mbps Ethernet, which could result in lost packets. The **Cortex** software is robust enough to deal gracefully with lost packets by ignoring the empty frames and continuing with the capture.

Another dataset in the Samples folder, **Eagles Face and Body**: Eagle one person, 60 Frames/sec, 400 Frames, 300-600 KBytes per camera, average maybe 450 KBytes (450 KBytes X 60 Frames/sec X 1/400 Frames = 68 KBytes /sec or about 680 Kbits/sec or about 0.7% of the 100 Mbps Ethernet).

For your 12 camera setup, this would be: 680 Kbps X 12 cameras = 8160 Kbps or about 8% of the 100 Mbps Ethernet.

Question	<i>Why is there is a problem displaying the AVI files when a 3D window is displayed on a Windows XP system? The AVI file plays in a jerky motion and the screen is sometimes sliced into horizontal blocks.</i>
Answer	This is a problem with Open GL and Direct X displaying at the same time. Right-click on your desktop, then select Properties > Settings > Advanced > Troubleshoot . Slide the Hardware Acceleration down a few points. Quit Cortex , then re-launch. See if that fixes the problem. The settings depend on what your graphics card does and has for features that changes from one computer to another. If that does not fix the problem, try to slide the Hardware Acceleration to None . Quit Cortex and then re-launch. If that still does not fix the problem, try to change some of the Open GL settings found under the name of your graphics card (found under the Properties > Setting > Advanced tab). Also, change the Vertical Sync setting to On by Default and restart your computer.
Question	<i>We came across a problem while in a two person w/ prop motion capture session. Whenever we recorded a motion, the recorded trc file would be missing a good portion of the marker data. Any idea why the data would just disappear?</i>
Answer	This is most likely cause by insufficient marker slots. The default marker slot setting is 192. In your case, you should increase the value. The marker slot setting maybe adjusted under Tools > Settings > Misc .
Question	<i>We are only using 93 markers (40 per actor, 5 per prop, and 3 for the ball). Shouldn't 192 marker slots be enough?</i>
Answer	Not necessarily. You should always have at least twice as many marker slots then actual markers. The number of slot is dependent on the tracking parameters settings. Each snippet of trajectory requires its own slot and different parameter values will create different sets of trajectories.
Question	<i>How will I know if I need more marker slots?</i>
Answer	After loading the file (trb/trc). In Post Processing Mode, scroll down the unnamed marker list (u_marker). If the all the slots are filled then you should increase the number of marker slots.
Question	<i>I just made some changes to my project file, but I do not want to recapture the entire motion list over again. Is there any way to rebuild the trb/trc data from the VC files?</i>
Answer	Yes. You can re-record the tracks files using the updated set of parameters in Cortex . First load the VC files. Then go to the Motion Capture > Output panel. Select to export a trb or trc file, you have to select the option OK to Overwrite if the tracks file already exists. Click on Record . The recording will automatically stop at the end of the VC file as long as the option "Loop Raw Files" is not checked (in the Tools > Settings > Playback tab). If it is checked then you will have to stop it by pressing the Stop button (same as the Record button, it changes name during the recording).

Question

Our templates in Cortex are always either extremely good or extremely poor. Can you give us some time so we can get a consistently good template each time?

Answer

The first step of the process is to collect both an init pose and a ROM(range of motion) for the actor. Do a manual identification of the makers in the init pose (T-pose or A-pose). Use this to create your template.

You should next be able to ID the first frame of the ROM (which should be a T-Pose) and do a Rectify through the whole ROM. Once the ROM has been completely identified, save the changes and then use Extend Template from the Create Template Dialog box.

The next step is to process all the easier motions. This way you can use these motions to extend your template further which will allow you to have a more complete template by the time you need to track the harder motion files.

Question

*I'm in the process of setting up **Cortex** to collect video data, and I have a question about compatible video devices. The manual indicates that a Sony camcorder with firewire output was used in testing. Would it be acceptable to run, for example, a surveillance camera through a capture device with a firewire output?*

Answer

Currently, Cortex looks for DV format streams over Firewire/IEEE1394 only, using those devices designed to operate with the Microsoft DV driver (MSDV).

A device delivering data over FireWire will likely use the MSDV driver if the data is in the 'DV' format, which refers to a specific compression codec (DV25), which is what most consumer handycams deliver.

WebCams, as a counter example, tend to deliver compressed Mpeg-2 streams. So it would not matter if the WebCam used USB or Firewire. If it delivers Mpeg-2 streams over Firewire, then it currently would not be recognized by EVaRT.

When evaluating a suitable camera, Firewire with DV25 is recommended.

More types of cameras are scheduled for inclusion into the "Reference Video" option in the future. Please check with support@motionanalysis.com for an updated list.

Appendix L *Useful Blank Forms*

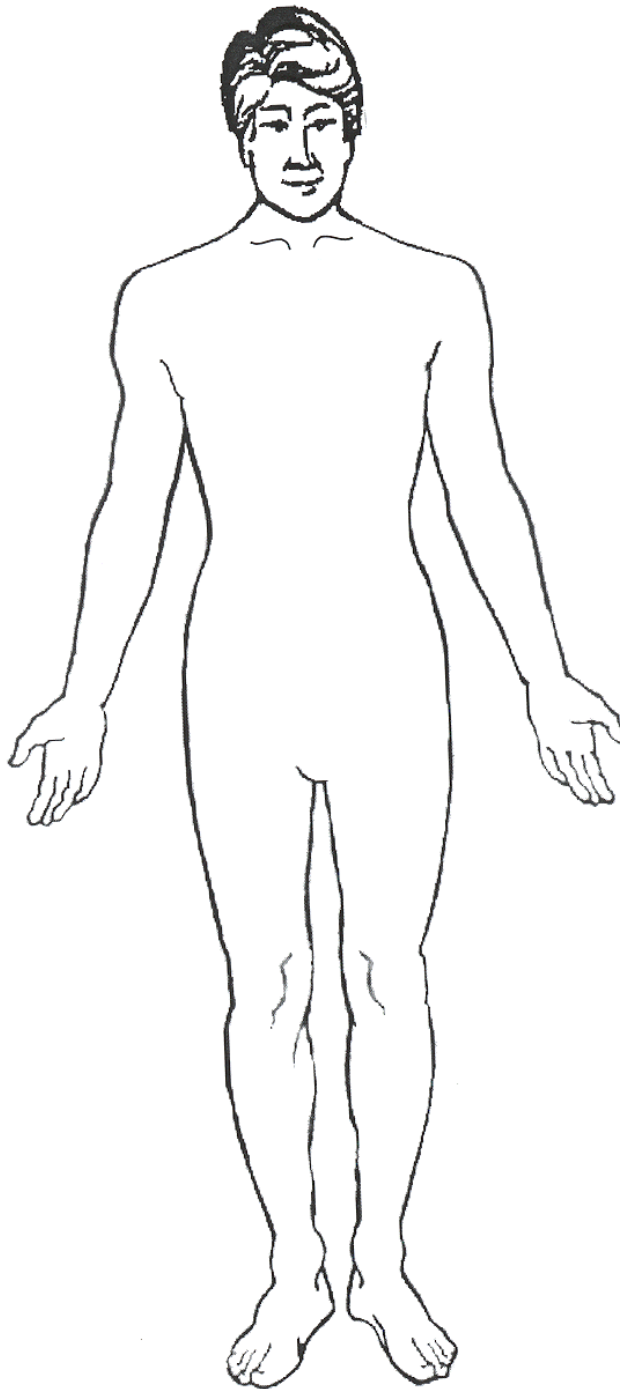
Topic	Page
Motion Capture Log	L-2
Human Body Outline—Front	L-3
Human Body Outline—Side	L-4
Human Body Outline—Back	L-5

The following blank forms may be useful to prepare for and document a motion capture session. Feel free to make copies as needed.

[illegible]

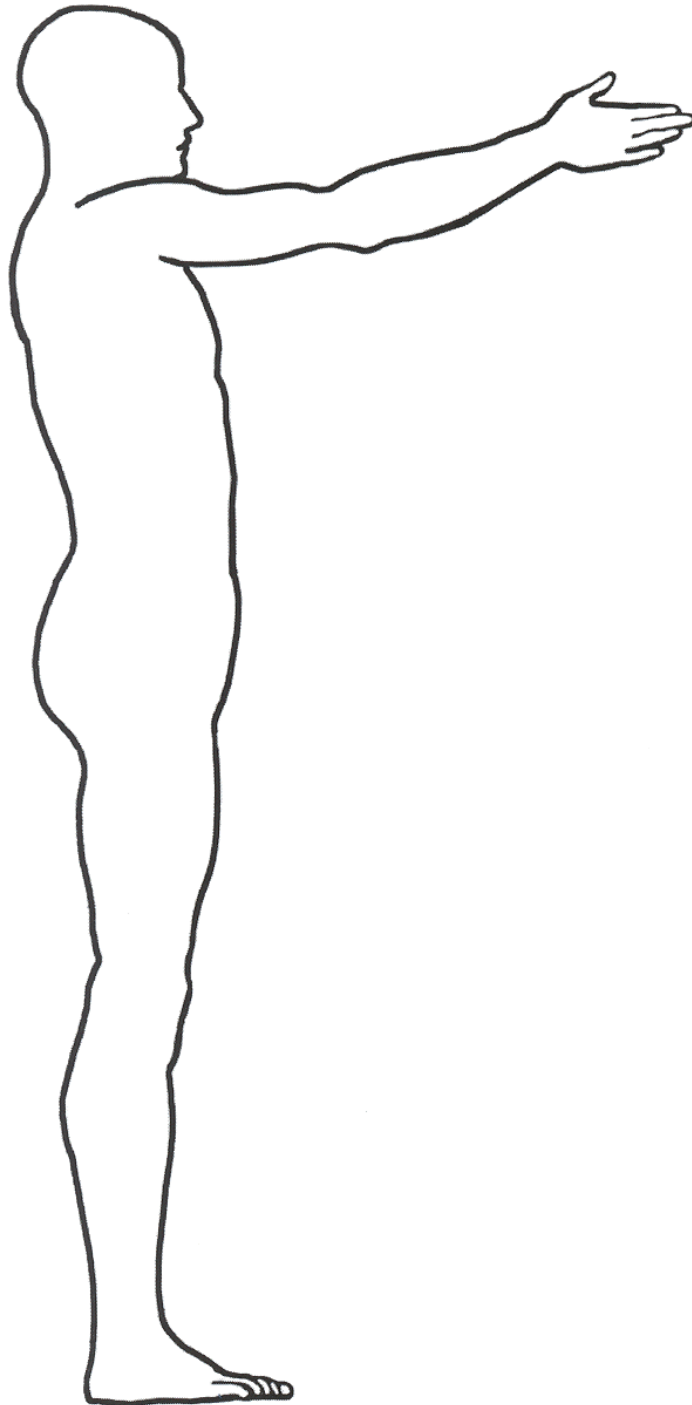
Human Body Outline—Front

Project _____ Date ____ / ____ / ____



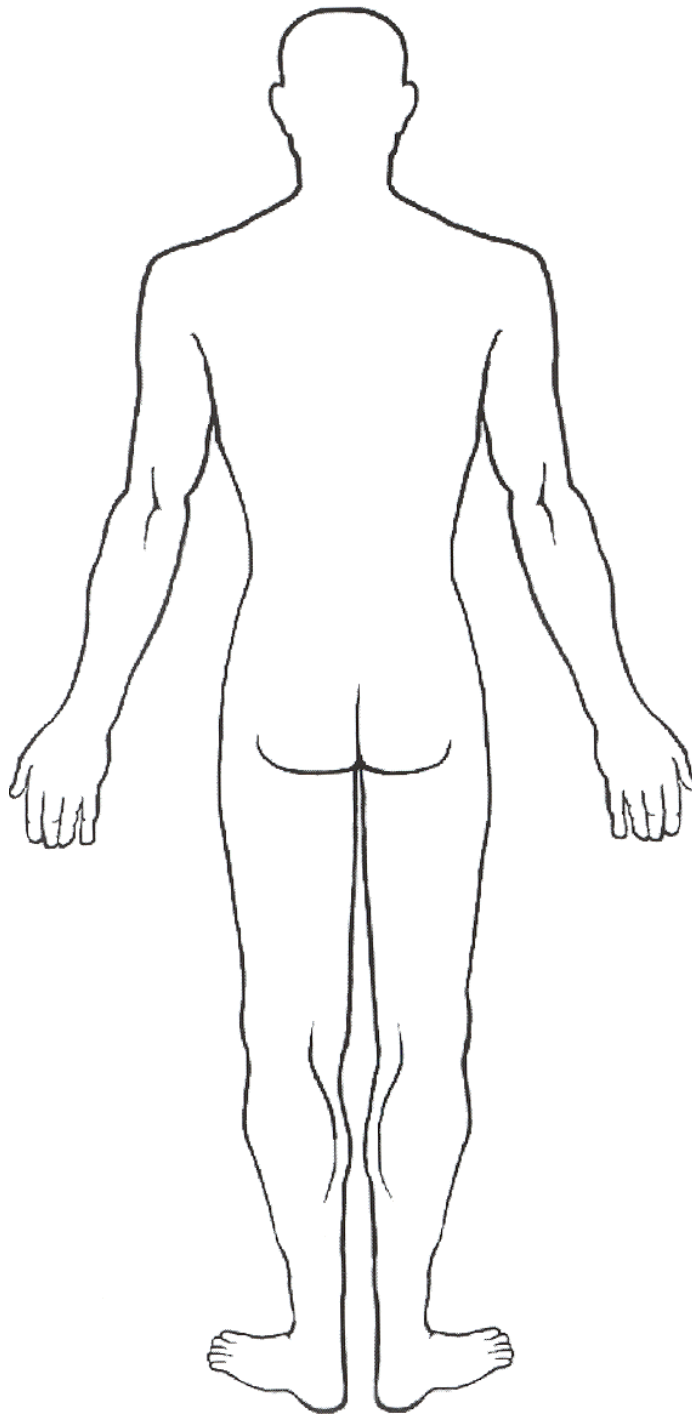
Human Body Outline—Side

Project _____ Date ____ / ____ / ____



Human Body Outline—Back

Project _____ Date ____/____/____



Index

Numerics

- 10 Camera Setup
 - Typical, 5-7
- 12 Camera Setup
 - Typical, 5-8
- 14 Camera Setup
 - Typical, 5-8
- 16 Camera Setup
 - Two-Tier, 5-4
 - Typical, 5-9
- 2 Markers, 11-17
- 2 Panes Top/Bottom, 6-32
- 28 Camera, 3-Tier Setup, 5-5
- 2D Display, 6-9, 6-31, 6-37
 - options, 6-37
 - pop-up menu, 6-37
- 3 Markers, 11-17
- 3 Point Average Filter, 10-10
- 32 Camera Setup
 - Typical, 5-10
- 3D Display Tab, 6-14
- 3D Studio Max, D-1
- 3D View
 - options, 6-34
 - pop-up menu, 6-34
- 4 Panes Layout, 6-31
- 5 Point Average Filter, 10-10
- 6 Camera Setup
 - Typical, 5-6
- 8 Camera Setup
 - Typical, 5-7
- 8 Eagle Camera Setup
 - Typical, 5-11

A

- Accuracy, 11-16
- Additional Tracking Objects, 9-9
- Adjusting Thresholds, 7-12
- Aiming Cameras, 8-9
- All Markers Radial Button, 10-15
- All On Button, 8-11
- Amplitude Zoom, 6-52
- AMTI Forceplates
 - Calibration Matrix, E-6
 - forcepla.cal, E-2
 - Using, E-6

- AMTI Gain Setting, E-6
- AMTI or Bertec Forceplates, B-4
- Analog Acquisition Rate, 7-16, B-13
- Analog ASCII Row Column, G-15
- Analog Channel Names
 - Replace, 6-26
- Analog Data Bit Depth, 6-7
- analog data files
 - opening, H-6
- Analog data graphs, 6-9
- Analog Display, 6-31, 6-38
- analog forceplate data, 6-31
- Analog Graphs
 - pop-up menu, 6-38
- Analog Input Channel Connections, B-12, B-14
- Analog Input Hardware Connections, B-1
 - Overview, B-1
- Analog Panel, 7-13
- Analog Setup, 7-13
- Analysis, J-1
 - Exporting Information, 10-32
 - graphs, 10-30
- Analysis Graphs F7, 6-10
- ANB, G-18
- ANB file, 6-31
- ANC, G-15
 - Example, G-15
- Animation, C-1
- Animation Plugins, 1-12
- Auto Scale, 10-4

B

- Backup Media, 4-4
- Bertec Forceplates
 - Calibration Matrix, E-6
 - forcepla.cal, E-2
 - Using, E-6
- Bertec Gain Setting, E-6
- Binary Files, G-18
 - ANB, G-18
 - C3D, G-18
 - TRB, G-18
- Biomechanics, C-3
- Blank Forms, L-1
 - Human Body Outline—Back, L-5
 - Human Body Outline—Front, L-3
 - Human Body Outline—Side, L-4
 - Motion Capture Log, L-2
- Building a Template, 9-5

Face, D-5
Butterworth Filter, 10-7

C

C3D, G-18
C3D files
 see tracked marker files
Calcium, 1-7
Calcium Segments, 11-15
Calculate Virtual Markers, 10-16
CalFloor.vc1, 8-14
Calibrate Panel, 8-3
calibrating
 system overview, 5-20
Calibrating Your System, 8-1
Calibration, 6-28
 Coordinate System, 5-20
 Face, 8-14
 Floor, 8-14
 from Previously Collected Files, 8-19
 Refining, 8-14
 Simulated, 8-19
Calibration Frame Tab, 8-5
Calibration L-Frame, 5-20
 Placing, 5-22
Calibration Panel, 8-1
Calibration Settings Window, 8-5
Calibration Settings Window Tabs, 8-5
 Capture Volume, 8-8
 Origin Offsets, 8-5
Calibration Tab, 6-11
Calibration Wand, 8-1
Calibration with Wand, 8-11
CalSeed.vcX, 8-10
CalWand.vcX, 8-13
Camcorder, 4-4
Camera Aiming, 8-4
Camera Aiming check box, 8-7
Camera Buttons, 6-49
 Right-Click, 6-49
camera buttons
 Real Time Dashboard, 6-41
Camera Problems, 5-24
Camera Software, 7-5
Cameras
 Optimum Number, 5-2
 Properly Seeded, 8-10
 Setting Up, 5-11
 Square (Seed) Calibration, 8-8

Cameras Panel, 7-1
Cameras Tab, 6-12
Capture Volume
 Sizes, 5-5
Capture Volume and Marker Size
 Relationship, 5-29
Capturing Facial Motion, D-1
 Accessory Kit, D-1
 Cheeks Marker Placement, D-5
 Chin Marker Placement, D-5
 Examples, D-7
 Eyebrows Marker Placement, D-4
 Eyelids Marker Placement, D-5
 Four Camera, D-3
 Head Marker Placement, D-4
 Jaw Marker Placement, D-5
 Lips Marker Placement, D-5
 Marker Placement, D-4
 Nose Bridge Marker Placement, D-5
 Nose Marker Placement, D-5
 Overview, D-1
 Setup, D-1
 System Configuration, D-1
 Three Camera, D-3
Centroid Parameters, 9-2
Channels Table, 6-38
Clear Entry, 9-9
Clear Marker Set Button, 11-3
Clearing Masks, 7-11
Collect and Calibrate Button, 8-10
Color Video, 6-9
Color Video option, 6-31
Colors, 6-25
Colors Form, 6-25
Connect Cameras Button, 6-48
Connect to Cameras, 7-1
Connections
 Analog Input Channel, B-12, B-14
 Analog Input Hardware, B-1
 EagleHub2, A-6
Control Points, 5-21
Convert .anc File..., 6-5
Coordinate System
 Calibration, 5-20
Copy, 10-18
CP-8 Power Hub
 Connections, A-5
Create .anb file, 6-7
Create .prj file, 6-7

Create .trc file, 6-7
 Create folder for files, 6-7
 Create forcepla.cal file, 6-7
 Create Linkages Button, 11-3
 Create Template, 9-6
 Creating Masks, 7-11
 Current Camera Information, 7-3
 Cut, 10-17
 Cut Inside, 10-17
 Cut Outside, 10-17

D

Data
 Viewing, 10-2
 Data Capture
 MA Quickstart, 2-10
 NM Quickstart, 3-8
 Data Painting, 10-28
 Data Views, 6-3, 6-9
 Dedicated Interface, 7-7
 Degrees of Freedom, 11-18
 Delete Artificial Data, 12-4
 Delete Outside Volume, 12-2
 Delete Short Snippets, 12-2
 Delete Unnamed, 10-19
 Digital Video Option, 6-40
 Director/Sequencer, 1-15
 Disconnect - Use Raw Files, 6-48
 Display Codes, 7-8
 Distance Between Two Markers Tab, 10-31
 Doc View, 6-10
 Dynamic Template Stretch Limits, 9-4

E

Eagle Camera
 Connections, A-6
 Dimensions, 5-17
 Display, 7-8
 Physical Dimensions, 5-17
 Power Consumption, A-3
 System Configuration, A-2
 EagleHub, ??-A-6
 EagleHub2 Connections, A-6
 Editing Tracked Data, 10-1
 EMG
 Muscle Name Selection, 7-14
 Signal Name Conventions, B-4
 Empty View, 6-10
 Enable COM1 Trigger, 9-14

Enable External Trigger, 9-13
 Ethernet Switch
 Connections, A-5
 Euler Angle Order, 13-7
 EVaDV Software, 1-11, 6-40
 EVaRT Project File, G-3
 EVaRT.ini, 6-34
 Examples
 Marker Sets, C-4
 Exchange, 10-17
 Export C3D File..., 6-5
 Export CRC File..., 6-5
 Export File Formats, G-1
 Export Forces File..., 6-5
 Export HTR File..., 6-4
 Export Kinetics File..., 6-5
 Export.ts (Time Series) File... button, 10-32
 Extend Template, 2-13
 Extending the Seed Calibration, 8-20
 Extra Stretch, 11-12
 Eye Movement, D-14

F

Face
 Building a Template, D-5
 Face Calibration, 8-14
 Facial Animation Techniques, D-9
 Falcon Camera
 Settings, 7-11
 FIFO slider, 6-31
 file management, 6-3
 Filters, 10-5
 Flash Drive
 Software Installation, 1-4
 Floor Calibration, 8-14
 Force Plate Channel Order, G-15
 forcepla.cal, E-1
 General Information, E-1
 Forcepla.cal File Format, E-1
 Forceplate
 3x3 Orientation Matrix, E-3
 Calibration Matrix, E-3
 File Data, E-3
 Number, E-3
 Optional Length & Width, E-3
 Scaling Factor, E-3
 True XYZ Origin, E-3
 XYZ Location in Video Coordinate System, E-3
 Forceplate Forces, 6-23

Forceplate Scaling Factor, E-5

Forceplates

6x6 Calibration Matrix, E-8

Frame Counter, 6-49

Frame Number, 10-3

Frame Offset, 12-5

Frame Rate, 7-2

Frames, Selecting, 10-5

G

gait analysis, 5-2

gait.lib, E-1

Gauss Newton, 11-15

Get marker names from desc, 6-7

ghost markers, K-1

Global Marker Data Adjustments, 12-4

Global Scale, 11-15

Going Live, 7-11

Graphics Only (Ctrl+G), 6-10

H

Hardware, 7-11

required, 1-3

Hardware Masks, 7-11

Hawk Camera

Connections, A-6

Dimensions, 5-18

Display Codes, 7-8

Physical Dimensions, 5-18

Power Consumption, A-3

System Configuration, A-2

Helen Hayes Marker Set, 2-9, C-3, C-4

Help Menu, 6-28

Hide Markers, 10-17

Hierarchical Translation & Rotation Data, 11-17

Hierarchical Translations and Rotations, G-5

High Frame Button, 10-3

Hinge Joints

Example, 11-20

Hot Keys and Tips, 6-51

HTR, G-9

HTR Version 1

Example, G-9

HTR Version 2

Example, G-10

HTR2, G-5

Example, G-6

I

ID Params Tab, 6-20

Identifying, 6-41

Identifying Markers, 10-12

Import File Formats, G-1, H-1, I-1, J-1

Included Angles Tab, 10-32

Information Center, 6-29

Installing Software, 1-4

International Society of Biomechanics, 5-21

IP Addresses, A-7

ISB, 5-21

J

Join, 10-24

Join Linear, 10-15

Join Virtual, 6-50, 10-24

Guidelines, 6-51, 10-27

Joining Gaps in Data, 10-5

K

Keyframe Animation, D-13

Kinematics and Kinetics, 6-5

Kinetics, 6-5

KinTrak, 1-15

Kistler Forceplates, B-4

Calibration Matrix, E-9

Example, E-9

forcepla.cal, E-2

Gain Setting, E-9

General Notes, E-10

Signal Names, E-9

True XYZ Origin, E-9

Using, E-9

Kyowa Dengyo Forceplates, E-10

L

Laboratory

Conditions, 5-1

Motion Capture, 5-1

Supplies, 5-1

layout control, 6-3

Lenses/Orientation, 8-7

Levenberg-Marquart, 11-16

Library Error, 5-29

License File

Sample, 1-5

Linkage Stretch Parameters, 6-21

Linkages

- Selecting, 6-52
- Links, 11-12
- Load Analog Setup..., 6-6
- Load Calibration..., 6-6
- Load Last Capture, 9-13, 9-14
- Load Marker Set..., 6-6
- Load Project, 6-31
- Load Tracks File, 6-32
- Loading New Camera Software, 7-5, 7-7
- Low and High Selected Frames, 10-4
- Low and High Visible Frames, 10-2

M

- mac_lic.dat, 1-5, G-2
- Main Marker Set, 9-9
- Make Unnamed, 10-16
- marker cloud
 - defined, H-1
- Marker Data Up Axis, 6-7
- Marker ID, 10-17
- Marker Name Prefix Options, 9-10
- Marker Placement
 - Capturing Facial Motion, D-4
 - Cheeks, D-5
 - Chin, D-5
 - Eyebrows, D-4
 - Eyelids, D-5
 - Head, D-4
 - Jaw, D-5
 - Lips, D-5
 - MA Quickstart, 2-8
 - NM Quickstart, 3-7
 - Nose, D-5
 - Nose Bridge, D-5
- Marker Sets, C-1
 - Animation, C-1
 - Biomechanics, C-3
 - Developing, C-5
 - Examples, C-4
 - Overview, C-1
- Marker Size, 5-29, 9-3
- Marker Slots, 6-18
- Marker XYZ Graphs, 6-9
- Markers
 - Identifying, 10-12
 - Selecting, 6-52
 - Unnamed, 10-11
- Markers Panel, 11-2
 - Clear Marker Set Button, 11-3

- Create Linkages Button, 11-3
- Masks
 - Clearing, 7-11
 - Creating, 7-11
- Matrix Method, 11-15
- Max Acceptable, 6-21, 9-4
- Max Horizontal Lines per Marker, 9-2
- Max Prediction Error, 9-3
- Max Residual, 9-3
- Max Speed (mm/frame), 10-23
- Max Target Speed, 9-3
- Max. Prediction Error (mm), 10-23
- Maya, D-1
- Menu Bar, 6-3
- Merge Marker Sets, 9-10
- Mesh Deformation, D-9
- Min. Cameras To Use, 9-3
- Min. Horizontal Lines per Marker, 9-2
- Misc Tab, 6-18
- mocap model, H-15
 - joint center calculations, H-13
 - marker set, H-14
 - scaling, H-13
 - static pose, H-10
- Mode Panel Buttons, 6-28
- Model Adjustments, 12-4
- Model Edit, 6-28, 11-1
 - Overview, 11-1
- Morphing, D-10
- Motion Analysis Corp.
 - Contact Information, 1-18
- Motion Analysis License File
 - Example, G-2
- Motion Capture, 6-28
 - Overview, 9-1
- Motion Capture Laboratory
 - Setting Up, 5-1
- Motion Capture of Hands, D-15
- Motion Capture Panel, 9-1
- Motion Capture Terminology, 4-5
- Motion Composer, 1-9
- Motion Module, H-1–H-17
 - opening analog data files, H-6
 - opening C3D files, H-2
 - opening TRB/C files, H-2
 - opening XLS files, H-6
 - real-time import, H-7
- Motionbuilder, D-1
- Move, 4-5

Move 1 Frame Button, 10-3
Move to Lowest/Highest Frame, 10-3
MTO, 9-7
Multiple Tracking Objects, 9-7
Muscle Name Selection
 EMG, 7-14

N

Network Configuration, A-7
Network Interface Card (NIC), A-7
New Subject Button, 6-43
New Subject button, 6-43
NI PCI-6071E, B-1
NI USB-6218, B-1, B-5
NIDAQ Software, B-3

O

OBJ, 11-7
Objects Panel, 9-8
Options to Change..., 10-19
Orient Body, 11-16
Origin Offsets Tab, 8-6
OrthoTrak, 1-14
 Intended Use, 1-14
Output Files, 9-13
Output Panel, 9-12

P

P3D, G-15
Panel Buttons, 6-29
Paste, 10-19
PCS_16Camera_2Tier.prj, 5-4
Phoneme Recognition, D-13
Pinging a camera, 5-27
Play Backward Button, 10-2
Play Speed, 10-3
Playback Tab, 6-13
Plugins, 6-28
Plugins Tab, 6-21, 12-1
pop-up menu, 6-34
 2D Display, 6-37
 3D Display, 6-34
 Analog Graphs, 6-38
 XYZ Graphs, 6-39
Pop-Up Menus, 6-34
Position, Velocity, and Acceleration Tab, 10-30
Positioning Cameras, 8-9
Post Process, 6-28

Post Process Dashboard, 6-50
Post Process mode, 6-32, 6-50
Post Process Tool Strip, 6-50, 10-12
Post Process Tools Tab, 6-15
Post Processing, 10-1
 Square Data, 8-21
 Strategies and Tips, 10-33
 Wand Data, 8-21
Post Trigger Mode, 9-13, 9-14
Power Consumption, A-3
 Eagle, A-3
 Hawk, A-3
PP Settings, 10-17, 10-19
PRJ Files, 6-30, G-3
Project Initialization
 MA Quickstart, 2-2
 NM Quickstart, 3-2
Prop Definition, 9-6
Props, 4-4
Protect Lens Correction, 6-18

Q

Questions and Answers, K-1
Quick ID, 10-17
quick solve
 defined, H-4
Quick-Start Tutorial
 Movement Analysis, 2-1

R

Range of Motion, 3-8, 9-5
Raw Files, 6-31
Raw Settings... Button, 6-48
Real Time Dashboard, 6-29, 6-40
 camera buttons, 6-41
real-time import
 see Motion Module
Record, 6-27
Recording Data, 9-14
Rectify, 10-18, 10-19
Rectify Functions, 10-19
Rectify Unnamed, 10-16, 10-20
Refine Tracks, 12-3
Refining a Calibration, 8-14
Refit Identifying Template, 12-4
Requirements
 System, 1-3
Reset IDs Button, 6-43
Rigid Body Join, 10-19

Rigid Body Rectify, 10-18, 10-19
 rigid objects, 9-9
 ROM, 3-8, 9-5
 rom.bin, 7-5
 Rotating, 6-51
 Rotation Offset, 11-14
 Run Button, 6-48
 functions, 6-48
 Run mode, 6-40

S

Sample Data
 Viewing, 6-31
 Sample Form, 4-5
 Samples directory, 6-31
 Save .ini Preferences..., 6-8
 Scaling, 10-4
 SDK, F-1
 SDK Streaming Options, 6-18
 Search, 10-17, 10-19
 Search Tab, 6-16
 Seed Calibration
 Extending, 8-20
 Segment
 Selecting, 6-52
 Select All Frames, 10-5
 Select Marker Set, 9-9
 Local, 9-9
 System Objects, 9-9
 Selected Markers Radial Button, 10-15
 Selecting Frames, 10-5
 Buttons, 10-3
 Selecting Linkages, 6-52
 Selecting Markers, 6-52
 Selecting Segments, 6-52
 Selecting Virtual Markers, 6-52
 Set as Master, 7-3
 Set Master Camera button, 7-3
 Setting Up
 Cameras, 5-11
 Settings
 Output Panel, 9-13
 Setup Analog
 MA Quickstart, 2-6
 Setup Panel, 7-1
 Shape Analysis, 9-3
 Show, 6-34, 8-17
 Show Camera Field of View, 5-12
 Show Camera Volume, 8-17
 Show Residuals and Cameras, 10-4
 Show Skin, 11-7
 Show Template Linkages, 10-13
 Shutter Speed, 7-2
 SIMM, 1-16
 Simulated Calibration, 8-19
 SkB, 1-8
 SkB Segments, 11-14
 Skeleton Builder, 1-8
 Skeleton Definitions, 6-5
 Skeleton Graphs, 6-10
 Skeleton Types, 13-1
 Skin File, 11-6
 Skin Transparency, 11-6
 Sky, 6-22
 Graphical User Interface, 14-2
 Sky Scripting Interface, 14-1
 Sky Writer, 14-1
 Slate Board, 4-4
 Smooth, 10-16
 Smoothing Tab, 6-15
 Snippets, 12-2
 Socket Error, 5-29
 SoftImage, D-1
 Software
 Installing, 1-4
 required, 1-4
 Software Developers Kit, F-1
 Sound Effects, 6-18
 Square (Seed) Calibration
 Cameras, 8-8
 Square Calibration, 8-2
 Standard System Configuration, A-2
 Starting EVaRT
 Quickstart, 2-1
 static pose
 see Motion Module
 Status Bar Messages, 6-29
 Still Camera, 4-4
 Streaming Options, 6-18
 Studio Preparation, 4-1
 Subject Preparation, 4-6
 Synchronizing Video, I-1, J-1
 System, 6-28
 System Calibrating Process, 5-20
 System Calibration
 MA Quickstart, 2-3
 NM Quickstart, 3-3
 System Hardware Interconnection, A-1

- Overview, A-1
- Standard System Configuration, A-2
- System Requirements, 1-3
- System Tab, 7-1

T

- Take, 4-5
- Talon Plugins, 1-13
- Target Marker, 10-4
- Template
 - Building, 9-5
- Template Create..., 10-13
- Template ID, 10-15
- Template ID Details, 10-20
- Template Rectify, 10-15, 10-19
- The, 10-17
- Threshold slider, 7-12
- Thresholds
 - Adjusting, 7-12
- Time Code, 6-18, 6-53, 10-3
- Time Code Counter, 6-49
- Time Lines, 6-22, 10-29
- Time Series Files, G-16
- Time Zoom
 - Method 1, 6-51
 - Method 2, 6-51
- Time Zoom Slider, 10-3
- To Reconsider, 6-20, 9-4
- Tool Menu, 6-30
- tools, 6-3
- Tools > Settings, 6-11
- Tools Menu, 6-11
- Track Row Column, G-3
- Tracked Data
 - Editing, 10-1
 - Viewing, 6-33
- tracked marker files
 - cropping ends, H-4
 - opening, H-2
- Tracking, 6-41
 - With More Than 8 Cameras, 5-14
- Tracking mode, 6-50
- Tracking Panel, 9-2
- Tracking Parameters, 9-2
- Translating, 6-51
- TRB, G-18
- TRB file, 6-31
- TRB/TRC files
 - see tracked marker files

- TRC, G-3
 - Data Header, G-4
 - Empty Fields, G-4
 - Example, G-4
 - File Header, G-3
 - Position Data, G-4
- Tree View Panel, 11-4
- Trial, 4-5
- Trim Capture W/Options..., 6-4
- tripod mounting points, 5-17, 5-18, 5-19
- Troubleshooting
 - Diagnosing Camera Problems, 5-24
 - Eagles and Hawks, 5-24
 - Ethernet, A-4
- TS, G-16
- Typical Camera Setup, 5-11

U

- Undo, 10-17
- Undo Tab, 6-17
- Unhide Markers, 10-18
- Uniform Scale, 10-4
- Unload Tracks Button, 6-55
- Unnamed Markers, 10-11
- Update Model, 12-4
- Update Model Pose, 12-4
- Update Model Pose Markers, 12-4
- Use Joint Limits, 11-16
- Use Per-Camera Settings, 9-3
- User Apps
 - Overview, 12-1, 13-1, 14-1
- User Interface, 6-1, 6-2

V

- VC file, 6-31
- Viewing Your Data, 10-2
- Virtual Join, 10-16
- Virtual Marker
 - Types, 11-17
- Virtual Marker Definitions, 6-23, 11-17
- Virtual Marker Tracks
 - Calculating, 11-21
- Virtual Markers, 11-17
 - Selecting, 6-52
- Visible Channels, 6-38
- Visible header, 6-38
- V-Marker Definitions Button, 11-21
- VMarkers, 11-11
- Volume, 5-13

W

- Waldos, D-14
- Wand, 8-1
- Wand Calibration, 8-2, 8-11
- Wand Calibration Coverage, 2-4, 3-5
- Wand Processing Status, 8-13
- Wand Processing Status Window
 - Accept Button, 8-13
 - Extend Seed Button, 8-13
 - Reject Button, 8-13
 - Run Again Button, 8-13
 - Stop Button, 8-13
- Web View, 6-10

X

- X Offset, 11-11
- X Panel, 12-2
- XYZ Graphs, 6-39
 - pop-up menu, 6-39

Y

- Y Offset, 11-11

Z

- Z Offset, 11-11
- Zoom
 - amplitude, 6-52
- Zoom In-Zoom Out, 10-4
- Zooming, 6-51
- Channel Settings, 6-7